

EXHIBIT 1

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC
Petitioner

v.

Jenam Tech, LLC
Patent Owner

Case IPR2021-00868
Patent No. 10,306,026

PATENT OWNER'S PRELIMINARY RESPONSE

Under 37 C.F.R. § 42.107

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
Submitted Electronically via PTAB E2E

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND	2
A.	The '026 Patent	2
B.	Person of Ordinary Skill in the Art (“POSITA”).....	4
C.	Overview of Prior Art	4
1.	<i>Wookey</i> (Ex1005).....	4
2.	<i>Berg</i> (Ex1007).....	5
III.	ARGUMENT	5
A.	Applicable Legal Principles.....	5
B.	This Petition Should Be Denied Under § 314(a) and/or § 325(d).....	7
1.	Factor One Favors Denial Because the Petition Was Filed Over a Year After Google Filed a Petition Against the Parent of the '026 Patent.....	12
2.	Petitioners Knew or Should Have Known About <i>Wookey</i> and <i>Berg</i> At the Time Google’s Earlier Petitions Were Filed.....	14
3.	Google Had Months to Analyze Jenam’s Responses and the Board’s Institution Decisions Concerning the '995 Patent Before Filing the Instant Petition.....	15
4.	Google Knew About <i>Wookey</i> and <i>Berg</i> When the First Petitions Were Filed and Well Before the Instant Petition was Filed	17
5.	Google Fails to Adequately Explain the Substantial Delay Between the Earlier Petitions Against the '995 Patent and the Instant Petition	17
6.	The Duplicative Follow-On Petition Challenging Substantially the Same Claims Would Strain the Board’s Resources.....	18

7.	Google’s Staggered, Duplicative Follow-On Petitions Make It Impossible for the Board to Consider The Petitions’ Evidence and Arguments in its Final Determination in the - 845 IPR	19
B.	Google Misinterprets the “Keep-Alive Period . . . Based on the Idle Time Period” Limitation.....	20
1.	The Claim’s Plain Language Requires a Distinct “Keep-Alive Period” and an “Idle Time Period”	21
2.	The Claim’s Plain Language Requires That the Term “Time Period” Denotes a Portion of Time, not a Value	24
C.	Contrary to Petitioner’s Misinterpretation, <i>Berg</i> Does Not Disclose the “Keep-alive Period . . . Based on the Idle Time Period” Limitation.....	25
D.	<i>Berg</i> Does Not Disclose the “Keep-alive Period . . . Based on the Idle Time Period” Limitation Even Under Petitioner’s Misinterpretation ...	30
E.	The Combination of <i>Wookey</i> , <i>Berg</i> , and/or <i>Tucker</i> Fails to Disclose the “Keep-alive Period . . . Based on the Idle Time Period” Limitation.....	33
1.	The combination of <i>Wookey</i> and <i>Berg</i> Would Not Have Rendered the ’026 Patent Claims Unpatentable	33
2.	The Combination of <i>Wookey</i> , <i>Berg</i> , and <i>Tucker</i> Would Not Have Rendered the ’026 Patent Claims Unpatentable.....	33
F.	This Redundant, Follow-On Petition Should Be Denied Under § 325(d) Because It Is Cumulative of the -742 IPR Petition.....	34
IV.	CONCLUSION.....	35

TABLE OF AUTHORITIES**Cases**

<i>Advanced Bionics, LLC v. MED-EL Elektromedizinische Geraete GmbH</i> , IPR2019-01469, Paper 6 (Feb. 13, 2020)	33
<i>Apple Inc. v. Contentguard Holdings, Inc.</i> , IPR2015-00442, Paper 9 (PTAB July 13, 2015)	6
<i>Apple Inc. v. Corephotonics Ltd.</i> , IPR2018-01356, Paper No. 9 (Feb. 5, 2019)	14, 17
<i>Becton, Dickinson & Co. v. Tyco Healthcare Grp., LP</i> , 616 F.3d 1249 (Fed. Cir. 2010)	23, 27
<i>Cat Tech LLC v. TubeMaster, Inc.</i> , 528 F.3d 871 (Fed. Cir. 2008)	23
<i>DISH Network LLC v. Broadband iTV, Inc.</i> , IPR2020-01281, Paper 17 (Feb. 4, 2021)	24
<i>Elekta Instrument S.A. v. O.U.R. Scientific Int’l, Inc.</i> , 214 F.3d 1302 (Fed. Cir. 2000)	23
<i>Ericsson Inc. v. Uniloc 2017 LLC</i> , IPR2020-00420, Paper 7 (PTAB June 18, 2020)	14
<i>Gaus v. Conair Corp.</i> , 363 F.3d 1284 (Fed. Cir. 2004)	23
<i>General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha</i> , IPR2016-01357, Paper 19 (PTAB Sept. 6, 2017)	passim
<i>Graham v. John Deere Co.</i> , 383 U.S. 1 (1966)	6
<i>Hisense Co., Ltd. et al v. Polaris PowerLED Techs., LLC</i> , IPR2020-01337, Paper 11 (Mar. 9, 2021)	24
<i>HTC Corp. v. Cellular Commc’ns Equip., LLC</i> , 701 F. App’x 978 (Fed. Cir. 2017)	23

<i>In re Magnum Oil Tools Int’l.</i> , 829 F.3d 1364 (Fed. Cir. 2016)	6
<i>In re Stepan Co.</i> , 868 F.3d 1342 (Fed. Cir. 2017)	6
<i>LONGi Green Energy Tech. Co. v. Hanwha Solutions Corp.</i> , IPR2020-00047, Paper 18.....	13, 14, 18
<i>Merck & Co. v. Teva Pharms. USA, Inc.</i> , 395 F.3d 1364 (Fed. Cir. 2005)	23
<i>NetApp Inc. v. Realtime Data, LLC</i> , IPR2017-01195, Paper 9 (PTAB Oct. 12, 2017).....	11
<i>Ortho-McNeil Pharm. v. Mylan Labs</i> , 520 F.3d 1358 (Fed. Cir. 2008)	6
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005)	21
<i>PNC Bank, NA v. Secure Axxess, LLC</i> , CBM2015-00039, Paper 9 (July 10, 2015).....	33
<i>SAS Institute Inc. v. Iancu</i> , 138 S. Ct. 1348 (2018).....	5
<i>Shenzhen Silver Star Intelligent Technology Co., Ltd. v. iRobot Corp.</i> , IPR2018-00761, Paper 1 (PTAB Sept. 5, 2018).....	11, 16
<i>Tietex Int’l, Ltd. v. Precision Fabrics Group, Inc.</i> , IPR2015-01671, Paper 7 (PTAB Feb. 11, 2016).....	7, 12, 13
<i>Unified Patents Inc. v. C-Cation Techs., LLC</i> , IPR2015-01045, Paper 15 (PTAB Oct. 7, 2015).....	6
<i>Unique Concepts, Inc. v. Brown</i> , 939 F.2d 1558 (Fed. Cir. 1991)	23
<i>United Fire Prot. Corp. v. Eng’rd Corrosion Sol’ns, LLC</i> , IPR2018-00991, Paper 9.....	16
<i>Vivint, Inc. v. Alarm.com Inc.</i> , 754 F. App’x 999 (Fed. Cir. 2018)	23

IPR2021-00868
Patent No. 10,306,026

Statutes

35 U.S.C. § 314.....	5, 7, 9, 20
35 U.S.C. § 316.....	10
35 U.S.C. §§ 314.....	10
37 C.F.R. § 42.107	1
37 C.F.R. § 42.108	9

Regulations

Trial Practice Guide Update, 83 Fed. Reg. 39,989	7
---	---

Patent Owner, Jenam Tech, LLC (“Jenam”) submits the following Patent Owner Preliminary Response pursuant to 37 C.F.R. § 42.107 (a). *See also* 37 C.F.R. § 42.107 (b).

I. INTRODUCTION

Petitioner, Google LLC (“Google”), filed a Petition for *Inter Partes* Review of claims 52, 55-57, 59-61, 64, and 66-69 of U.S. Patent No. 10,306,026 (“the ’026 patent”, Ex. 1001) on April 29, 2021.¹ (Paper No. 1 (“Petition”).) Jenam disclaimed claims 52, 61, 68, and 69. Only claims 55-57, 59-60, 64, and 66-67 remain. Because the asserted references fail to disclose every limitation of claims 55 and 64, and their dependent claims, Google has not shown that there is a reasonable likelihood that it will prevail. The Petition should therefore be denied.

As an initial matter, this Petition is one of ten petitions that Google has filed against the ’026 patent and its family members. The multiple petitions rely on overlapping prior art references including references cited in another petition filed by Unified Patents, LLC. These staggered, numerous petitions pose a substantial burden for Jenam and are designed to give Petitioner an unfair advantage by

¹ This Petition is one of ten that Google filed against the ’026 patent and its family members. Additionally, another IPR is pending involving a related patent using overlapping prior art that was filed by Unified Patents, LLC (“Unified”).

allowing it to preview Jenam's arguments regarding the primary prior art references being relied upon. This vexatious behavior should not be condoned.

Even on the merits, the petition still fails. For example, *Berg* does not disclose the limitation "one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period," recited in claims 55 and 64, and the remaining claims which depend from claims 55 and 64.

II. BACKGROUND

A. The '026 Patent

The '026 patent discloses apparatuses and methods for sharing information between nodes on a network to detect idle connections using TCP, as well as TCP-variants and non-TCP, to improve network performance. (*See e.g.*, Figs. 6 and 7 of '026 patent, reproduced below.) The '026 patent teaches, for example, that when a second node receives a packet in a connection, a portion of the packet is detected as identifying metadata for a period that may be detectable by a first node. Based on the metadata for the period, the second node modifies a timeout attribute that is associated with the connection. (Ex. 1001, 4:67-5:3.) In addition, the '026 teaches that a packet may be generated that includes a portion identifying metadata for the period based on information for detecting a period received at the first node. This generated packet may then be sent to the second node. (Ex. 1001, Abstract, 4:66-5:7.)

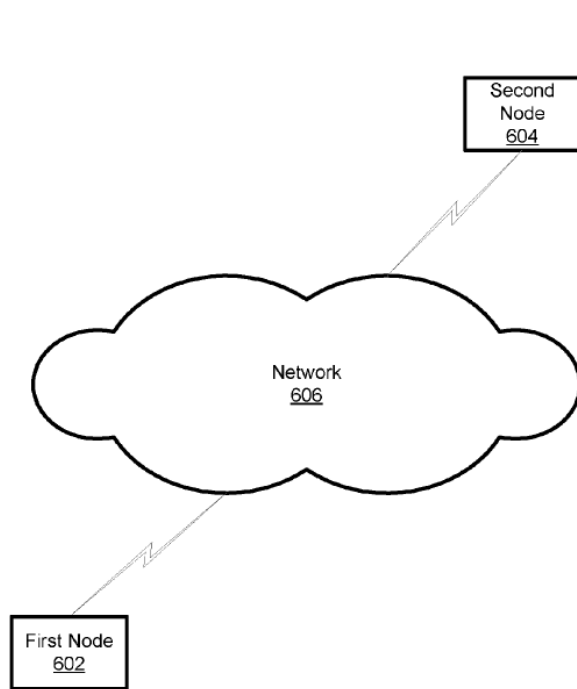


Fig. 6

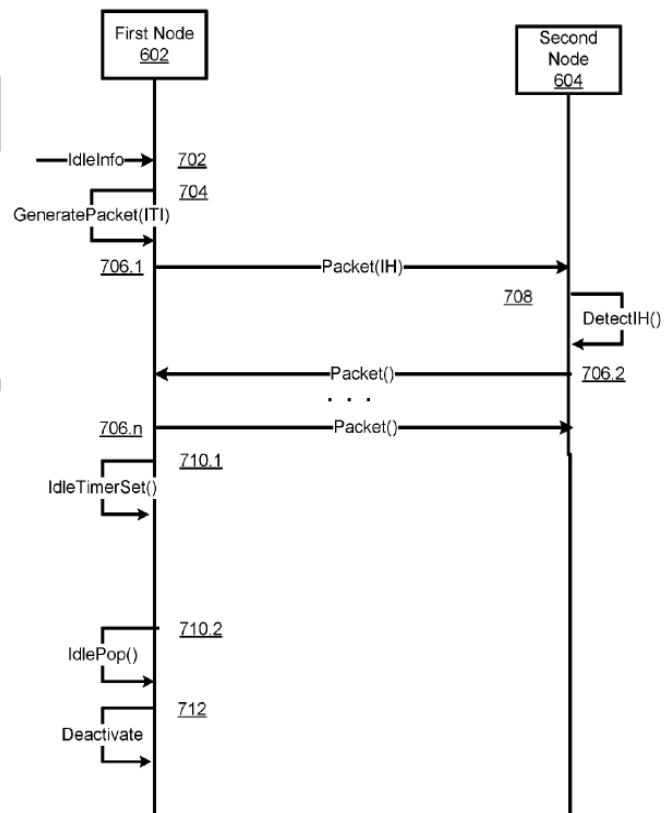


Fig. 7

In one example, the '026 patent discloses a method in which a node causes a data packet to be sent to another node, the latter node receives the packet, and detects an idle time period parameter field, identifies metadata in the idle time period parameter field for an idle time period, during which no packet is communicated in a TCP-variant connection to keep the connection active, and modifies a timeout attribute based on the metadata. (EX. 1001, 3:4-11.) This is also reflected in independent claims 52 and 61. (Ex. 1001, 27:37-57; 28:27-48.) Further, claims 55 and 64, which depend from claims 52 and 61 respectively,

contain an additional limitation reciting that “one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.” Claims 56-57 and 59-60 depend from claim 55; and claims 66-67 depend from claim 64, at least in part, and thus contain the foregoing limitations.

In another example, a node receives idle information for detecting an idle time period during which no packet is communicated in a transmission control protocol-variant connection to keep the TCP-variant connection active, generates a packet including an idle time period parameter field and sends the packet to another node in advance of the TCP-variant connection is established for use by the another node in modifying a timeout attribute associated with the TCP-variant connection. (Ex. 1001, 4:45-61.)

One or more of these aspects of the invention are included in the challenged claims. (Ex1001.) They are not present in the asserted references.

B. Person of Ordinary Skill in the Art (“POSITA”)

For the purposes of this POPR, Jenam agrees with Petitioner’s definition of a POSITA.

C. Overview of Prior Art

1. *Wookey* (Ex1005)

Wookey is a voluminous reference disclosing numerous disparate embodiments across its 241 pages, including over 100 different figures and around

250 columns of disclosures having over 1100 paragraphs. (Ex1005.) It generally relates to accessing resources on a network.

2. *Berg* (Ex1007)

Berg teaches a method and apparatus for providing continuous voice and call communications between a data network and a telephony network. Specifically, in one configuration, a first gateway controller that receives the call communications from the data network is in communication with multiple sessions to a gateway that couples the communications to the telephony network. If an in-use session fails, the communications are switched from the failed session to another session, automatically.

III. ARGUMENT

Google has not demonstrated that a POSITA would have found the challenged claims of the '026 patent obvious by a reasonable likelihood in view of *Wookey* and *Berg* or in further view of *Tucker*. Its arguments rely on incorrect interpretations of the claim and asserted references. As set forth herein, the Petition should be denied under 35 U.S.C. § 314. Moreover, should the Board disagree, it should still deny the Petition pursuant to its discretionary authority.

A. Applicable Legal Principles

Google is required to show that it is likely to prevail as to at least one claim of the '026 patent. 35 U.S.C. § 314; *SAS Institute Inc. v. Iancu*, 138 S. Ct. 1348,

1353 (2018). To make a prima facie showing of obviousness, a petition must fulfill the requirements of *Graham v. John Deere Co.*, 383 U.S. 1 (1966), including demonstrating that the cited references disclose **each element** of a challenged claim. (*Id.*) *In re Magnum Oil Tools Int’l.*, 829 F.3d 1364, 1376 (Fed. Cir. 2016); *Apple Inc. v. Contentguard Holdings, Inc.*, IPR2015-00442, Paper 9 at 12-13 (PTAB July 13, 2015).

Google must also show there would be some **motivation to combine** the asserted references in the manner asserted, and that a POSITA would have had a **reasonable expectation of success** in doing so. *In re Stepan Co.*, 868 F.3d 1342, 1345-46 (Fed. Cir. 2017). Further, even if individual modifications would have been obvious, a petition must explain why making **all** of the changes would have been obvious. *Apple Inc. v. Contentguard*, Paper 9 at 16-17.

Hindsight analysis is inappropriate. Obviousness must be measured “**at the time the invention was made.**” *Ortho-McNeil Pharm. v. Mylan Labs*, 520 F.3d 1358, 1364 (Fed. Cir. 2008) (emphasis in original).²

Even where a petition meets the statutory threshold, the decision whether to institute is discretionary. *Unified Patents Inc. v. C-Cation Techs., LLC*, IPR2015-01045, Paper 15 at 3 (PTAB Oct. 7, 2015). The Board may deny a petition based

² Unless indicated otherwise, all emphasis in this POPR is added.

on “the potential impacts on both the efficiency of the [IPR] process and the fundamental fairness of the process for all parties.” Trial Practice Guide Update, 83 Fed. Reg. 39,989, at 9-11 (August 13, 2008); *see also Comcast Cable Comm’ns, LLC v. Rovi Guides, Inc.*, IPR2019-00279, Paper 7 at 3 (April 22, 2019)

B. This Petition Should Be Denied Under § 314(a) and/or § 325(d)

This Petition and Google’s multiple other follow-on petitions for the ’026 patent and its family members should be denied. The Board should exercise its discretion under 35 U.S.C. § 314(a) to deny institution of this and the concurrently filed petitions attacking the ’945 and ’026 patents because institution of a trial on any of these proceedings would promote neither the efficiency nor the fundamental fairness of the IPR process for all parties. To the contrary, instituting this or any of the other petitions would reward Petitioner’s behavior, which is designed to harass Jenam by forcing it to defend against numerous similar, serial challenges to related patents.

The Board has previously denied institution of such IPRs filed against related patents because it placed an unfair burden on the patent owner. *Tietex Int’l, Ltd. v. Precision Fabrics Group, Inc.*, IPR2015-01671, Paper 7, at 8 (PTAB Feb. 11, 2016). Similarly, in *Comcast Cable Communications, LLC, v. Rovi Guides, Inc.*, IPR2019-00279, IPR2019-00280, IPR2019-00281, IPR2019-00282,

IPR2021-00868
Patent No. 10,306,026

IPR2019-00283, Paper 7 at 3 (April 22, 2019), the Board acknowledged that “when there are other related patents also each challenged by multiple petitions at the same time . . . this can undermine the Office’s ability to complete proceedings in a timely manner and may place an unfair burden on the Patent Owner.”

This Petition is *one of two* that Google filed against the ’026 patent. It is also *one of four* that Google filed on April 29, 2021 against related patents owned by Jenam. Google previously filed *four petitions* for *inter partes* review on March 15, 2021 against two other related patents owned by Jenam, and filed a petition for post-grant review on May 10, 2021 for yet another related patent. The following table summarizes the various petitions, challenged patents and claims, and asserted references, and illustrates the *substantial overlap* that exists across the various proceedings.

Petitioner	IPR No. & Filing Date	Challenged Patent ³ & Claim(s)	Asserted References
Google	-845 IPR, filed 4/17/2020	’995 patent: claims 1-24	<i>Eggert, Hankinson and RFC 1122</i>
Google	-627 IPR, filed 3/15/2021	’215 patent: claims 1, 4, 8 and 9	<i>Wookey, Eggert and Abdolbaghian</i>

³ All challenged patents are related.

IPR2021-00868
Patent No. 10,306,026

Google	-628 IPR, 3/15/2021	'026 patent: claims 16, 22 and 23	<i>Wookey</i> and <i>Eggert</i>
Google	-629 IPR, 3/15/2021	'026 patent: claims 24-29	<i>Wookey</i> and <i>Berg</i>
Google	-630 IPR, 3/15/2021	'026 patent: claims 25 and 28	<i>Wookey</i> and <i>Berg</i>
Google	-867 IPR, filed 4/29/2021	'026 patent: claims 1, 17, 21- 22, 31-33, 43, 45-48 and 51	<i>Eggert</i> , <i>Abdolbaghian</i> and <i>Tucker</i>
Google	-868 IPR, filed 4/29/2021	'026 patent: claims 52, 55-57, 59-61, 64, 66-69	<i>Wookey</i> , <i>Berg</i> and <i>Tucker</i>
Google	-869 IPR, filed 4/29/2021	'945 patent: claims 1, 9-10, 34 and 64-69	<i>Wookey</i> and <i>Berg</i>
Google	-870 IPR, filed 4/29/2021	'945 patent: claims 7-8 and 11-12	<i>Wookey</i> , <i>Berg</i> and <i>Eggert</i>
Google	-082 PGR, filed 5/10/2021	'774 patent: claim 1	<i>Wookey</i> and <i>Eggert</i>

Because Google's petitions challenge related patents that cover the same subject matter and include overlapping claim limitations, these follow-on petitions place an unfair burden on Patent Owner while providing a tactical advantage to Petitioner, and should therefore be denied.

The Director has discretion to deny an IPR under 35 U.S.C. § 314(a) and 37

C.F.R. § 42.108(a). *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19, at 5-19 (PTAB Sept. 6, 2017) (“*General Plastic*”); Consolidated Trial Practice Guide, November 2019 (“Practice Guide”) at 55. The Board has consistently considered a number of non-exhaustive factors in determining whether to exercise that discretion. *General Plastic* at 15-19 (listing and discussing factors). Indeed, the Practice Guide states:

The *General Plastic* factors are also not exclusive and are not intended to represent all situations where it may be appropriate to deny a petition. *Id.* at 16. There may be other reasons besides the “follow-on” petition context where the “effect . . . on the economy, the integrity of the patent system, the efficient administration of the Office, and the ability of the Office to timely complete proceedings,” 35 U.S.C. § 316(b), favors denying a petition even though some claims meet the threshold standards for institution under 35 U.S.C. §§ 314(a), and 324(a).

Practice Guide at 58.

This Petition represents a redundant challenge that, for the reasons described below, should be denied. Here, the earlier proceedings filed by Google—the -845, -627, -628, -629, and -630 IPRs—involve related patents and the same parties. This Petition therefore presents a textbook case of “abuse of the review process by repeated attacks on patents,” and should be denied. *General Plastic* at 17 (citing Congressional records that “[IPR is] not to be used as tools for harassment . . . through repeated litigation and administrative attacks on the

validity of a patent. Doing so would frustrate the purpose of the section as providing quick and cost effective alternatives to litigation.”).

Indeed, where, as here, a later-filed petition is filed after receipt of a preliminary response or a decision on institution, at least one PTAB Judge has noted that “there is a rebuttable presumption” that the later-filed petition will be denied. *Shenzhen Silver Star Intelligent Technology Co., Ltd. v. iRobot Corp.*, IPR2018-00761, Paper 15 at 16-17 (PTAB Sept. 5, 2018). The multiple follow-on petitions on related patents give Google an unfair advantage, allowing it to see Jenam’s responses in the earlier proceedings involving the ’995 patent, with similar references and similar arguments, and allowing Google to craft multiple different arguments and responses in the follow-on proceedings. The Board has cautioned against such “gamesmanship.” *Id.* at 19.

When “the circumstances show that the availability of Patent Owner’s arguments . . . provided substantial potential benefit to Petitioner to tailor its arguments,” *NetApp Inc. v. Realtime Data, LLC*, IPR2017-01195, Paper 9, 11 n.12 (PTAB Oct. 12, 2017) (“*NetApp*”), the Board has exercised discretion to deny institution because the Board recognizes “the potential for abuse of the review process by repeated attacks on patents,” and intends to “take undue inequities and prejudices to Patent Owner into account.” *General Plastic* at 17.

IPR2021-00868
Patent No. 10,306,026

Here, because institution of a trial on any of these proceedings would promote neither the efficiency of the IPR process nor the fundamental fairness of the process for all parties, the Petition should be denied. Instituting this or any of the other petitions would reward Petitioner's behavior, which is designed to harass Jenam by forcing it to defend against similar, serial challenges to the same or related patents. The Board has previously denied institution of such IPRs filed against related patents because it placed an unfair burden on the patent owner. *Tietex Int'l, Ltd. v. Precision Fabrics Group, Inc.*, IPR2015-01671, Paper 7, at 8 (PTAB Feb. 11, 2016). Similarly, in *Comcast Cable Communications, LLC, v. Rovi Guides, Inc.*, IPR2019-00279, IPR2019-00280, IPR2019-00281, IPR2019-00282, IPR2019-00283, Paper 7 at 3 (April 22, 2019), the Board acknowledged that "when there are other related patents also each challenged by multiple petitions at the same time . . . this can undermine the Office's ability to complete proceedings in a timely manner and may place an unfair burden on the Patent Owner."

For these reasons and those discussed in more detail below, this untimely, follow-on Petition should be denied.

1. Factor One Favors Denial Because the Petition Was Filed Over a Year After Google Filed a Petition Against the Parent of the '026 Patent

The first factor favors denial. The *first General Plastic* factor inquires “whether the same petitioner previously filed a petition directed to the same claims of the same patent.” *Id.* at 16. Here, Google’s first petition, filed nearly a year prior to the filing of the Petition here, targeted the claims of the ’995 patent. The ’995 patent is the parent of the ’026 patent challenged in this proceeding. Accordingly, the ’995 and ’026 patents share a common specification and the challenged claims of the ’995 patent and claims 55-57, 59-60, 64, and 66-67 at issue here include many overlapping limitations.

As set forth above, the Board has previously denied institution when related, follow-on petitions challenged claims similar in scope and content, even when the claims were in different patents. *Tietex International, LTD., v. Precision Fabrics Group, Inc.*, IPR2015-01671, Paper 7, at 8 (PTAB Feb. 11, 2016) (“Notably, in this proceeding, the claims of the ’162 patent are so similar in scope and content to those in the ’639.”). The Board should similarly exercise its discretion to deny this Petition. Otherwise Jenam would be forced to expend resources defending claims that are similar in scope and content to those of the ’995 patent it is already defending. *Id.* at 9 (“In particular, we are not persuaded that Patent Owner should be forced to expend resources defending the claims in the ’162 patent.”); *LONGi Green Energy Tech. Co. v. Hanwha Solutions Corp.*, IPR2020-00047, Paper 18

(“*LONGi*”), at 8 (discretionary denial despite the second petition being directed to different subset of claims of the same patent).

2. Petitioners Knew or Should Have Known About *Wookey* and *Berg* At the Time Google’s Earlier Petitions Were Filed

Factor two also weighs against institution of the Petition. The *second General Plastic* factor inquires “whether at the time of filing the first petition the petitioner knew of the prior art asserted in the second petition or should have known of it.” *General Plastic* at 16.

Google does not indicate if *Wookey* was available or unavailable at the time of filing the first petition. The Board has found that if the Petitioner does not indicate if a reference was previously available, this factor weighs in favor of denying the petition. *Apple Inc. v. Corephotonics Ltd.*, IPR2018-01356, Paper No. 9, at 7 (Feb. 5, 2019) (“Petitioner does not indicate whether Iwasaki was available or unavailable at the time of filing the Petition in the ’1146 IPR . . . Thus, factors 1 and 2 weigh in favor of exercising our discretion to deny institution.”); *LONGi* at 9-10 (finding factor 2 favored discretionary denial because “Petitioner either knew or should have known of [the prior art]”); *Ericsson Inc. v. Uniloc 2017 LLC*, IPR2020-00420, Paper 7 at 10-11 (PTAB June 18, 2020) (finding that petitioner’s failure to represent it had no prior knowledge of references weighed in favor of denial). Regardless, *Wookey* was cited in numerous Google patent applications

prior to the filing of the '845 IPR. (*E.g.*, U.S. Patent Nos. 9,092,767 (“’767 patent”; Ex2006); 9,489,240 (“’240 patent”; Ex2007); 9,858,572 (“’572 patent”; Ex2008); 10,185,954 (“’954 patent”; Ex2009).) Thus, Google *was aware of Wookey* when it filed the '845 IPR.

Accordingly, the second *General Plastic* factor favors denial.

3. Google Had Months to Analyze Jenam’s Responses and the Board’s Institution Decisions Concerning the '995 Patent Before Filing the Instant Petition

The *third General Plastic* factor asks “whether at the time of filing the second petition the petitioner had already received the [POPR] to the first petition or received the Board’s decision on [institution].” *General Plastic* at 16. Here, at the time of filing the follow-on petitions (including the instant Petition), Google had already received the POPR, the Board’s institution decision, and the Patent Owner’s Response (“POR”) in the -845 IPR,⁴ and made ample use of them in preparing the second-filed petitions. Google additionally had the benefit on Jenam’s infringement contentions. Google took advantage of these documents and revised its invalidity challenges in the instant Petition.

⁴ The Patent Owner’s Response in the -845 IPR was filed on December 31, 2020.

This Petition was filed on April 29, 2021.

The Board has stated that factor 3 “is directed to Petitioner’s potential benefit from receiving and having the opportunity to study Patent Owner’s Preliminary Response, as well as our institution decisions on the first-filed petitions, prior to its filing of follow-on petitions.” *General Plastic* at 17. The Board has found factor 3 dispositive and strongly influential when “the circumstances show that the availability of Patent Owner’s arguments . . . provided substantial potential benefit to Petitioner to tailor its arguments.” *United Fire Prot. Corp. v. Eng’rd Corrosion Sol’ns, LLC*, IPR2018-00991, Paper 9 at 15; *NetApp* at 11 n.12 (“Factor 3 is directed to situations in which a petitioner delays filing a subsequent petition so that it can tailor its arguments to address issues identified by the patent owner and/or the Board during a prior proceeding.”).

When a Petitioner waits to file its Petition until after having time to review Patent Owner’s arguments, there is no other conclusion other than it was for tactical reasons. *Shenzhen Silver Star Intelligent Technology Co., Ltd. v. iRobot Corp.*, IPR2018-0761, Paper 15 at 19 (PTAB Sept. 5, 2018). (“[A]bsent explanation, we have no other conclusion to make but that the second petition was filed at the time it was filed solely for the tactical reason of using the first petition as a test case.”).

Accordingly, *General Plastic* factor 3 weighs strongly in favor of denial.

4. Google Knew About *Wookey* and *Berg* When the First Petitions Were Filed and Well Before the Instant Petition was Filed

The *fourth General Plastic* factor concerns “the length of time that elapsed between the time the petitioner learned of the prior art asserted in the second petition and the filing of the second petition.” *General Plastic* at 16. Here, as set forth above, Google was likely aware of *Wookey* at the time of filing the petition in the -845 IPR in April 2020. Even if it was not, this factor nevertheless weighs in favor of denying the Petition because of Google’s failure to address this issue, *Apple*, IPR2018-01356, Paper No. 9, at 7, and because multiple Google patent applications cited *Wookey* during prosecution well before any of the petitions discussed herein were filed (*E.g.*, U.S. Patent Nos. 9,092,767 (“’767 patent”; Ex2006); 9,489,240 (“’240 patent”; Ex2007); 9,858,572 (“’572 patent”; Ex2008); 10,185,954 (“’954 patent”; Ex2009).)

Similarly, as set forth above, Google should have been aware of *Berg* at the time the -745 IPR petition was filed in April 2020.

5. Google Fails to Adequately Explain the Substantial Delay Between the Earlier Petitions Against the ’995 Patent and the Instant Petition

The *fifth General Plastic* factor asks “whether the petitioner provides adequate explanation for the time elapsed between the filings of multiple petitions directed to the same claims of the same patent.” *General Plastic* at 16. Here,

Google has not explained the time elapsed between the filings of multiple petitions directed to the related patents based on substantially the same prior art references that it was aware of when the first petitions were filed against the '995 patent. To the extent a reasonable explanation exists for Google's delay, it was incumbent upon Google to identify those circumstances in its Petition. Thus, factor 5 weighs in favor of exercising discretion to deny institution of the present proceeding. *See Apple*, IPR2018-01356, Paper No. 9 at 8.

6. The Duplicative Follow-On Petition Challenging Substantially the Same Claims Would Strain the Board's Resources

The *sixth General Plastic* factor concerns "the finite resources of the Board." *General Plastic* at 16. Here, the Board would be required to expend extensive resources to adjudicate one petition against the parent '995 patent, four petitions challenging related patents simultaneously filed on March 15, 2021, and four more petitions, including the instant Petition (which is one of two challenging the '026 patent), filed on April 29, 2021. The Board may also be required to expend significant resources adjudicating the one petition for PGR filed on May 20, 2021 against another related patent.

Given the delay of more than six months between the filings of the first and the two waves of follow-on petitions, the Board could not feasibly coordinate or consolidate the proceedings, but rather will have to adjudicate many of the same

claim limitations based on the same prior art. Such redundancy would be an inefficient use of the Board's limited resources. "In general, having multiple petitions challenging the same patent . . . is inefficient and tends to waste resources." *Valve*, at 15. Accordingly, *General Plastic* factor six militates in favor of denial. *LONGi*, at 12 ("Moving forward with separate proceedings involving overlapping issues, but having different evidentiary records and schedules, would have a significant impact on the Board's resources").

7. Google's Staggered, Duplicative Follow-On Petitions Make It Impossible for the Board to Consider The Petitions' Evidence and Arguments in its Final Determination in the - 845 IPR

The *seventh* *General Plastic* factor concerns the requirement to "issue a final determination not later than 1 year after the date on which the Director notices institution of review." *General Plastic* at 16. Here, the Board has already instituted review of the challenged claims of the '995 patent. IPR2020-00845, Paper 16 (Oct. 8, 2020). Patent Owner submitted the Patent Owner's Response (Paper 18) on December 31, 2020. Oral argument was held on July 8, 2021. If the Board seeks to evaluate additional evidence and arguments from the newly submitted four petitions targeting substantially the same claim limitations as in the '995 patent, the Board would not be able to reach a final determination in the

first IPR, namely, IPR2020-00845, by October 8, 2021, as required. Accordingly, *General Plastic* factor seven also militates in favor of discretionary denial.

Based on the unexplained delay in filing the last wave of four follow-on petitions (including the instant Petition), the substantial overlap with the first-filed petitions in IPR2020-00845 and IPR2020-00742, and the overlap with the four follow-on petitions filed on March 15, 2021, and in view of the Board's rationale in *General Plastic*, the Board should exercise its discretion under Section 314(a) to deny the instant Petition.

Accordingly, for at least the reasons discussed above, the evidence shows that the *General Plastic* factors weigh against institution. As a result, the Petition should be denied pursuant to the Board's discretion under 35 U.S.C § 314(a).

B. Google Misinterprets the “Keep-Alive Period . . . Based on the Idle Time Period” Limitation

Google and its expert have applied improper claim interpretations in asserting invalidity of the challenged claims. While the claims should be given their ordinary and customary meaning, Google and its expert have twisted the meaning to something very different from ordinary. As set forth below, when the claims are properly understood to be given their (actual) ordinary and customary meaning, Google's arguments must be rejected. Specifically, Google's apparent

interpretation of the “keep-alive period . . . based on the idle time period”

limitation in claims 55 and 64 is incorrect and should be rejected.

Claim Language	Jenam’s Proposed Construction	Google’s Misinterpretation
“one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, <i>based on</i> the idle time period” ⁵	“one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period which is an <u>actual separate period in relation to the keep-alive period</u> ”	“one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period where the keep-alive period and the idle time period are capable of referring to a <u>same</u> period or aspects thereof ”

1. The Claim’s Plain Language Requires a Distinct “Keep-Alive Period” and an “Idle Time Period”

Under the governing claim construction standard, words of the claim must be given their plain meaning. *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). Here, the patentee chose to use multiple distinct terms to define the scope of this claim limitation, one of which is “based on” another. The phrase “based on” is a plain-English term that, in the context of two nouns (“X” & “Y”) where X is “based on” Y, a POSITA would understand to mean that Y serves or forms a basis for X.⁶ The patentee gave no special definitions in the specification

⁵ Unless indicated otherwise, all emphasis in this POPR is added.

⁶ <https://www.merriam-webster.com/dictionary/base>

or prosecution history for this plain-English phrase. Thus, by its plain language, the claim requires an idle time period which is a separate time period, in relation to the keep-alive period which is based on such idle time period.

The '026 patent specification confirms that the claimed “keep-alive period” and “idle time period” are distinct periods. For instance, the specification repeatedly refers to a “keep-alive timeout” attribute (with an associated period “duration”), determined by a “keep-alive timer” which may expire before an “idle time period” can occur:

In another aspect, first node 602 may set a *keep-alive timeout* attribute based on a duration of the first *idle time period* identified in the first idle information and/or in the metadata provided to second node 604. For example, first node 602 may monitor a *time period* during which no non-empty packets are sent or received in the TCP connection. A keep-alive option handler and/or keep-alive component (not shown) operating in first node 602 may set a *keep-alive timer* according to the *timeout attribute*, with a *duration* that will result in the *keep-alive timer* expiring before an *idle time period* can occur. In response to detecting a *keep-alive timeout*, which may be indicated by the expiration of the keep-alive timer, the keep-alive option handler and/or keep-alive policy component may provide information to packet generator component 552 to generate a TCP keep-alive packet.

(Ex1001 at 16:43-58.)

As the excerpt above shows, the “idle timeout” value may be identified in the “idle information and/or in the metadata provided to” the second node. (*Id.* at 16:45-46.) On the other hand, a duration of the “keep-alive timeout” period is determined by a “keep-alive timer,” which in turn is set by a “keep-alive option handler and/or keep-alive component.” (*Id.* at 16:49-51.) Petitioner cites ***nothing*** in the specification to support its apparent interpretation that the “keep-alive timeout” period and the “idle time period” refer to the same “period.” Indeed, the specification does not disclose a single example where the “keep-alive timeout” period and the “idle time period” are synonymous. Petitioner’s apparent interpretation would therefore contradict the plain language of the claims and the specification and is improper. *Vivint, Inc. v. Alarm.com Inc.*, 754 F. App’x 999, 1004-05 (Fed. Cir. 2018) (reversing the Board’s erroneous construction of a term as it suggested the opposite of what the patent discussed). Moreover, equating the two limitations would have the effect of rendering the intervening phrase “based on” superfluous, which is disfavored. *See Cat Tech LLC v. TubeMaster, Inc.*, 528 F.3d 871, 885 (Fed. Cir. 2008) (refusing to adopt a claim construction that would render a claim limitation meaningless) (citing *Unique Concepts, Inc. v. Brown*, 939 F.2d 1558, 1563 (Fed. Cir. 1991)); *Elekta Instrument S.A. v. O.U.R. Scientific Int’l, Inc.*, 214 F.3d 1302, 1307 (Fed. Cir. 2000) (construing a claim term to avoid rendering a claim limitation superfluous); *Merck & Co. v. Teva Pharms. USA, Inc.*,

395 F.3d 1364, 1372 (Fed. Cir. 2005) (rejecting a proposed claim construction that would render “other parts of the claim superfluous.”).

Moreover, “[w]here a claim lists elements separately, ‘the clear implication of the claim language’ is that those elements are ‘distinct component[s]’ of the patented invention.” *Becton, Dickinson & Co. v. Tyco Healthcare Grp., LP*, 616 F.3d 1249, 1254 (Fed. Cir. 2010) (quoting *Gaus v. Conair Corp.*, 363 F.3d 1284, 1288 (Fed. Cir. 2004)); *see also HTC Corp. v. Cellular Commc’ns Equip., LLC*, 701 F. App’x 978, 982 (Fed. Cir. 2017). The Board has repeatedly followed this principle. *See, e.g., Hisense Co., Ltd. et al v. Polaris PowerLED Techs., LLC*, IPR2020-01337, Paper 11, at 18 (Mar. 9, 2021); *DISH Network LLC v. Broadband iTV, Inc.*, IPR2020-01281, Paper 17, at 19 (Feb. 4, 2021); *Comcast Cable Commc’ns, LLC v. Rovi Guides, Inc.*, IPR2017-00951, Paper 42, at 17 (Sep. 9, 2018). Accordingly, here, the “keep-alive period” and “idle time period” are “distinct components” of the claimed inventions and cannot represent the same “period” (or aspects thereof).

2. The Claim’s Plain Language Requires That the Term “Time Period” Denotes a Portion of Time, not a Value

Claim 52, from which claim 55 depends, recites “metadata, that specifies at least one of a number of seconds or minutes, in an idle time period parameter field . . . for an idle time period” and “determining, based on the metadata, a timeout

attribute.” Similarly, claim 61, from which claim 64 depends, recites “an idle time period parameter field identifying metadata that is specified in at least one of a number of seconds or minutes.”

The '026 patent specification similarly confirms that the claimed “keep-alive period” and “idle time period” are portions of time, not values. Specifically, the specification repeatedly uses “time period” in the phrase “time period, during which” (something does or does not occur) distinctly and separately from the disclosed metadata *value* specified in seconds or minutes. (Ex. 1001 at 16:43-58 (“first node 602 may monitor a *time period during which* no non-empty packets are sent or received in the TCP connection”).) The metadata value may thus *represent* a length of a time period, but is *not* synonymous with the time period itself. Stated differently, events may (or may not) occur during a time “period”; but nothing happens during a value—“during a value” does not make sense.

Thus, the “idle time period” and “keep alive period” limitations both refer to portions of time, and neither limitation refers to the value representing the duration of any time period. Indeed, the '026 patent specification does not include *any* teaching where a “time period” can refer to a metadata “value” of that same “period.”

C. Contrary to Petitioner’s Misinterpretation, *Berg* Does Not Disclose the “Keep-alive Period . . . Based on the Idle Time Period” Limitation

The Petition should be denied because the disclosures Petitioner points to in the *Wookey-Berg* combination do not satisfy the “keep-alive period . . . based on the idle time period” limitation of claims 55 and 64.

As set forth above, the claimed “keep-alive period” and “idle time period” ***cannot*** refer to the same time period, as such a misinterpretation would contradict the plain language of the claims and render the phrase “based on” superfluous. However, in the Petition, Petitioner equates the two limitations in precisely such a manner.

Specifically, Petitioner makes it clear that they rely on the “null segment timeout value” in the “parameter field in an RUDP SYN segment” to meet Patent Owner’s claimed “idle time period.” This can be inferred from Petitioner’s specific reference to the ‘**null segment timeout value (“idle time period”)** parameter field in an RUDP SYN segment’ (emphasis added). (Pet. at 37) Petitioner then contends that the “duration” of the same “null segment timer” meets the “keep-alive period” limitation of claim 55. (Pet. at 45, citing Ex. 1007 at 20:36-38, 23:46-59.)

Petitioner thus clearly relies on the same “period” in *Berg* to meet both the “keep-alive period” and the “idle time period” limitations, and even explicitly acknowledges this fact. For instance, Petitioner first argues that the “idle time period” is the “amount of time that a client will wait . . . before sending a null

segment.” (Pet. at 37-38.) Petitioner asserts later in the Petition that the keep-alive period is also the null segment timer’s “duration,” where “the client will send a null segment . . . to the server when its null segment timer expires.” (*Id.* at 45.)

Petitioner’s argument is thus best summarized in the table below:

Claim Limitations	<i>Berg</i> Disclosure
“idle time period”	null segment timeout “ value ” <i>used to set</i> “duration” of “ null segment timeout ” period
“keep-alive period”	“ duration ” of the same “ null segment timeout ” period

In asserting that *both* the “keep-alive period” and “idle time period” limitations are satisfied by aspects of the *same* “null segment timer” in *Berg*, Petitioner thus misinterprets the claims to allow or require that the “idle time period” and “keep-alive period” limitations both refer to the *same* period. As set forth above, however, Petitioner’s misinterpretation is inconsistent with the fundamental claim construction principle that the two limitations are “distinct component[s],” and must be rejected. *Becton*, 616 F.3d at 1254.

Petitioner attempts to overcome this fact by arguing that the duration for the null segment timer is “set proportionally” to—and is therefore “based on”—the null segment timeout value. (Pet. at 45.) However, Petitioner fails to cite any evidence in *Berg* for its assertion that a duration is “proportional” to a timeout value, or that the “duration” is anything other than the “timeout value” itself.

Indeed, the words “proportional” or “proportionally” do not appear anywhere in *Berg*, and *Berg* even clarifies, to the contrary, that the “Null Segment Timeout Value” *is* the parameter value (specified in milliseconds) for the timeout. (Ex. 1007 at 20:36-42.)

Petitioner’s related argument that the null segment timer’s duration “mirrors” the idle time period, (Pet. at 46), similarly fails. *Berg* does not disclose any “mirror[ing]” whatsoever. *Berg* teaches only one “period”—the Null Segment Timeout” period—whose “value” and “duration” are merely two aspects of it, and not, in and of themselves, separate “time periods.”

Petitioner therefore fails to identify *two* separate periods in *Berg* to meet the separately claimed “keep-alive period” and “idle time period” limitations of claims 55 and 64. To this end, *Berg* does not even suggest, let alone teach, “one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period,” as the claims require.

Petitioner’s misinterpretation of the claims does not stop here. Petitioner *also* relies on *Berg*’s *same* null segment timeout “value” to *also* meet the claimed “metadata” [‘null segment timeout value (“metadata”)’ Pet. At 45]. Petitioner’s argument is thus best summarized in the table below:

PO’s Claims	Berg Disclosure
-------------	-----------------

“idle time period”	null segment timeout “ value ” used to set “duration” of “ <u>NULL SEGMENT TIMEOUT</u> ” period
“metadata”	<u>SAME</u> : null segment timeout “ value ” used to set “duration” of “ <u>NULL SEGMENT TIMEOUT</u> ” period

This line of argument constitutes yet *another* example of how Petitioner improperly relies on a *same* entity in *Berg* to meet *different* claim elements, which is clearly improper for the reasons discussed above.

Petitioner’s analysis for claims 56 and 64 is further evidence of the absurdity of its position. Claims 56 and 64 further recite that “the keep-alive period is not configurable, while the idle time period is configurable.” Here, Petitioner argues simply that “the null segment timeout value (metadata specifying the ‘idle time period’) is configurable, while the null segment timer’s duration (‘keep-alive period’) is not.” (Pet. at 47.) Petitioner is technically and logically wrong. The timeout “value” that Petitioner relies on to meet the “idle time period” is not a “period,” but a “value” that is, in turn, used to configure the “null segment timer’s duration.” Thus, if the “value” is “configurable,” the duration that is set *based on* the same metadata “value” is also configurable.

Petitioner attempts to gloss over this inconsistency by referring to values at different times during the configuration process as meeting the “idle time period”

and “keep alive period” limitations. For instance, Petitioner argues that the “idle time period” in the *Wookey-Berg* combination is configurable “because the client can negotiate the null segment timeout value,” (Pet. at 47), but goes on to assert that “*once the connection is established*,” “the client’s null segment timer’s duration is set to a fixed proportion of the period specified by the null segment timeout value.” (*Id.* at 48.) Even if the client’s null segment timer’s duration is fixed at a certain time during the operation of the device in *Berg*, this fact is irrelevant. The claim limitation requires that “the keep-alive period is *not* configurable,” not that it is “not configurable during an arbitrary interval of time.” Here, Petitioner admits that the null segment timeout value, which the null segment time period duration is *based on*, is configurable. Therefore, by Petitioner’s own admission and its mapping of the claim limitations onto the null segment timeout value and time period duration respectively, *Berg* does not include any teaching where “the keep-alive period is not configurable, while the idle time period is configurable.”

Petitioner’s argument as to claims 56 and 64 thus amounts to no more than a temporal sleight of hand, which is unsupported (and even contradicted) by the claim language and ’026 patent’s specification.

D. *Berg* Does Not Disclose the “Keep-alive Period . . . Based on the Idle Time Period” Limitation Even Under Petitioner’s Misinterpretation

Even if the Board views the challenged claims in light of Petitioner's misinterpretation of this limitation as requiring (or permitting) that the "keep-alive period" be separate from the "idle time period," Petitioner has not established that *Berg* teaches it, for the reasons set forth below.

First, even assuming that the claimed "keep alive period" and "idle time period" are coextensive (they are not), Petitioner's reliance on parameters for different aspects of the same "null segment timeout" to satisfy the two separately-claimed "periods" is improper. As set forth above, Petitioner asserts that the "idle time period" is met by the "null segment timeout" **value** in *Berg*, while the "keep-alive period" is met by the "**duration**" of the "null segment timer." (Pet. at 45.) However, Petitioner is not free to rely on both a "value" and a "duration" of the **same** "time period" in *Berg* to satisfy the two, separately-claimed "time period" limitations. Regardless, irrespective of whether the Board adopts Patent Owner's proposed construction, the null segment timeout "value" that Petitioner relies on to meet the "idle time period" is not, itself, a "period," but rather merely an admitted metadata "value" that is used to configure a duration of the actual "null segment timeout" period. Indeed, as set forth above, Petitioner relies on the same null segment timeout "value" to **also** meet the claimed "metadata" limitation (which constitutes **yet another** example of how Petitioner improperly relies on a same feature in *Berg* to meet **different** claim elements).

Moreover, neither a “metadata” nor a “value” is a time “period,” but is merely something that describes an aspect (e.g., a duration) of the actual time “period.” Thus, *Berg* does not even suggest “one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period,” as required by claims 55 and 64.

In parallel proceedings, which do not involve claims reciting a “keep-alive period” limitation, Petitioner understood the above fact and did not improperly rely on *Berg*’s metadata “value” to meet the claimed “idle time period” limitation. Instead, as evidenced by the excerpts below, Petitioner argued that *Berg*’s “Null Segment Timeout” itself met the claimed “idle time period” limitation:

“As discussed below, the *Wookey-Berg* combination discloses identifying a null segment timeout value (‘metadata’) in the null segment timeout parameter field for a *null segment timeout (‘idle time period’)*, where, after the null segment timeout is detected, the RUDP connection is deemed inactive.”

(Page 48 of petition in IPR2021-00869.)

“This *waiting period (when the connection is idle)* before the client sends a null segment to the server to verify whether the connection is active represents an *‘idle time period.’*”

(Pages 49-50 of petition in IPR2021-00869.)

“The duration of the client’s null segment timer is set to the null segment timeout (‘idle time period’) specified by the null segment timeout value. That is, the null segment timeout value defines a duration of time that the connection will remain inactive before the client sends a null segment to see if the connection still exists.”

(Page 50 of petition in IPR2021-00869.)

Petitioner’s shifting sands theory of invalidity based on *Berg* further evidences the fact that Petitioner’s arguments are merely concocted with a goal of meeting the claimed limitations. As *Berg* does not disclose the “metadata,” “idle time period,” and “keep alive period” limitations, as the challenged claims require, the Petition should be denied.

E. The Combination of *Wookey*, *Berg*, and/or *Tucker* Fails to Disclose the “Keep-alive Period . . . Based on the Idle Time Period” Limitation

1. The combination of *Wookey* and *Berg* Would Not Have Rendered the ’026 Patent Claims Unpatentable

As explained above, *Berg* does not disclose the methods of claims 55 and/or claim 64, which the remaining claims depend from. *Wookey* does not cure these deficiencies, and Petitioner does not contend it does. Therefore, Ground 1 fails.

2. The Combination of *Wookey*, *Berg*, and *Tucker* Would Not Have Rendered the ’026 Patent Claims Unpatentable

As explained above, Wookey does not disclose the methods of claims 55 and/or claim 64. *Tucker* does not cure these deficiencies, and Petitioner does not contend it does. Therefore, Ground 2 fails.

F. This Redundant, Follow-On Petition Should Be Denied Under § 325(d) Because It Is Cumulative of the -742 IPR Petition

As discussed above, § 325(d) allows the Board to deny institution if the challenge is based on prior art and/or arguments previously presented to the Office. *Advanced Bionics, LLC v. MED-EL Elektromedizinische Geraete GmbH*, IPR2019-01469, Paper 6, at 7-8 (Feb. 13, 2020). This includes *art* and *arguments* provided in AIA post-grant proceedings. *Id.* at 8.

Google has not met its burden to show that the prior art and arguments in this proceeding are not cumulative of prior art or arguments previously presented to the Office. *See PNC Bank, NA v. Secure Access, LLC*, CBM2015-00039, Paper 9, at 19 (July 10, 2015). Indeed, Google does not attempt to distinguish the instant Petition's use of *Berg* with its use in the -742 IPR. Nor could it. The arguments relating to *Berg* in the instant Petition are substantially similar to those set forth in the -742 IPR. For at least these reasons, the Petition should be denied.

IPR2021-00868
Patent No. 10,306,026

IV. CONCLUSION

For the foregoing reasons, the Petition to institute *inter partes* review of 55-57, 59-60, 64, and 66-67 of the '026 patent should be denied.

IPR2021-00868
Patent No. 10,306,026

Dated: August 13, 2021

DEVLIN LAW FIRM LLC

/ Timothy Devlin /

Timothy Devlin (Reg. No. 41,706)

Derek Dahlgren (*pro hac vice*
pending)

1526 Gilpin Avenue

Wilmington, DE 19806

(302)-449-9010

tdevlin@devlinlawfirm.com

ddahlgren@devlinlawfirm.com

Attorneys for Patent Owner

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that there are 7,483 words in this paper, excluding any table of contents, table of authorities, mandatory notices under 37 C.F.R. § 42.8, certificate of word count, certificate of service, or appendix of exhibits. This certification relies on the word count of the word-processing system used to prepare this paper.

Respectfully submitted,

/ Timothy Devlin /

Timothy Devlin
Registration No. 41,706
DEVLIN LAW FIRM LLC
1306 N. Broom Street, First Floor
Wilmington, DE 19806
(302)-449-9010
tdevlin@devlinlawfirm.com
Attorney for Patent Owner

CERTIFICATE OF SERVICE

I hereby certify that on August 13, 2021, I caused a true and correct copy of the foregoing materials to be served via the Patent Office electronic filing system, and electronic service via email to the following attorneys of record pursuant to Petitioners' consent.

<u>LEAD COUNSEL</u>	<u>BACK-UP COUNSEL</u>
Naveen Modi (Reg. 46,224) Paul Hastings LLP 2050 M Street NW Washington, D.C. 20036 Tel: (202) 551-1700	Joseph E. Palys (Reg. 46,508) Quadeer A. Ahmed (Reg. 60,835) Jason Heidemann (Reg. 77,880) Paul Hastings LLP 2050 M Street NW Washington, D.C. 20036 Tel: (202) 551-1700
Additional email for service: PH-Google-Jenam-IPR@paulhastings.com	

/ Timothy Devlin/
 Timothy Devlin

EXHIBIT 2

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC, LG ELECTRONICS, INC.,
LG ELECTRONICS U.S.A., INC.,
Petitioners

v.

JENAM TECH, LLC
Patent Owner

Case No. IPR2020-00845

U.S. Patent No. 9,923,995

**PETITION FOR INTER PARTES REVIEW
OF U.S. PATENT NO. 9,923,995**

Patent No. 9,923,995
Petition for *Inter Partes* Review

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. TECHNOLOGY BACKGROUND.....	1
III. THE '995 PATENT.....	6
A. Overview	6
B. Prosecution History	9
IV. PERSON OF ORDINARY SKILL IN THE ART	10
V. CLAIM CONSTRUCTION	10
VI. GROUNDS FORMING THE BASIS OF THIS PETITION	11
VII. THE PRIOR ART.....	12
A. Eggert	12
1. Eggert is a “Printed Publication”	12
2. Overview of Eggert.....	14
B. Hankinson.....	15
C. RFC 1122	17
VIII. CLAIMS 1-24 ARE UNPATENTABLE UNDER §103	18
A. Ground 1: Claims 1-24 Would Have Been Obvious Over Eggert.....	18
1. Claim 1/15.....	18
a) 1(pre)/15(pre) An apparatus comprising:	18
b) 1(a)/15(a) a non-transitory memory storing instructions; and.....	18
c) 1(b)/15(b) one or more processors in communication with the non-transitory memory,	19
d) 1(c)/15(c) wherein the one or more processors execute the instructions for:.....	19

Patent No. 9,923,995

Petition for *Inter Partes* Review**TABLE OF CONTENTS**

(continued)

	Page
e) 1(d) receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established;.....	20
f) 1(d) (TCP)-variant packet, TCP-variant connection;.....	23
g) 1(e) detecting an idle time period parameter field in the TCP-variant packet;.....	26
h) 1(f) identifying metadata in the idle time period parameter field for an idle time period and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active	29
i) 1(g) modifying, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection.	32
2. Claim 2	35
3. Claim 3	36
a) 3(a) The apparatus of claim 1 wherein at least one of: the second node includes a server, the server being configured to: in response to the receiving, send, by the second node to the first node, another TCP-variant packet in advance of the TCP-variant connection being established, the another TCP-variant packet including other metadata for the idle time period;.....	36

Patent No. 9,923,995

Petition for *Inter Partes* Review**TABLE OF CONTENTS**

(continued)

	Page
b) 3(b) the second node includes a client, the client being configured such that the receiving is performed subsequent to sending, by the second node to the first node, another TCP-variant packet in advance of the TCP-variant connection being established including other metadata for the idle time period.	37
4. Claims 4 and 5.....	38
5. Claims 6/19	39
6. Claim 7	39
7. Claim 8	40
8. Claim 9	41
9. Claims 10/20	41
10. Claim 11/21	42
a) [11.a]/[21.a]	42
b) [11.b]/[21.b].....	43
11. Claim 12/22.....	45
a) [12.a].....	45
b) [12.b].....	45
12. Claims 13/23	46
a) [13.a]/[23.a]	46
b) [13.b].....	48
c) [23.b].....	48
d) [13.c]/[23.c]	49
e) [13.d]/[23.d].....	50
13. Claim 14.....	50
14. Claim 15.....	51

Patent No. 9,923,995
Petition for *Inter Partes* Review

TABLE OF CONTENTS
(continued)

	Page
a) 15(d) receiving idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active:	51
b) 15(e) generating a TCP-variant packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information; and.....	54
c) 15(f) sending, from a first node to a second node, the TCP-variant packet in advance of the TCP-variant connection being established to provide the metadata for the idle time period to the second node,.....	56
d) 15(g) for use by the second node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.	57
15. Claim 16.....	58
16. Claim 17.....	59
17. Claim 18.....	59
a) [18.a].....	59
b) [18.b].....	60
18. Claim 24.....	61
B. Ground 2: Claims 1-24 Would Have Been Obvious Over Eggert In View Of Hankinson	62
1. Claims 1/15	63
a) 1(a)/15(a) a non-transitory memory storing instructions; and.....	64
b) 1(b)/15(b) one or more processors in communication with the non-transitory memory,	65

Patent No. 9,923,995

Petition for *Inter Partes* Review**TABLE OF CONTENTS**

(continued)

	Page
2. Claims 11/21	66
3. Claim 12(b)	67
4. Claims 13/23	68
C. Ground 3: Claim 16 Would Have Been Obvious Over Eggert In View Of Hankinson and Further In View Of RFC 1122	68
IX. DISCRETIONARY DENIAL UNDER 35 U.S.C. §325(d) WOULD BE INAPPROPRIATE	69
A. The Prior Art Relied On Herein Is Not The Same Or Substantially Similar To Any Art The Examiner Considered	69
B. The Prior Art Relied On Herein Is Not The Same Or Substantially Similar To Any Art In Unified's IPR Petition	72
X. DISCRETIONARY DENIAL UNDER 35 U.S.C. §314(a) WOULD BE INAPPROPRIATE	73
A. <i>Unified</i> Provides No Basis For Discretionary Denial	73
B. The Related Litigation Provides No Basis For Discretionary Denial	74
XI. MANDATORY NOTICES PURSUANT TO 37 C.F.R. §42.8	76
A. Real Parties-In-Interest under 37 C.F.R. §42.8(b)(1).	76
B. Related Matters Under 37 C.F.R. §42.8(b)(2).	77
C. Lead and Back-up Counsel under 37 C.F.R. §42.8(b)(3).	78
D. Service Information under 37 C.F.R. §42.8(b)(4).	78
XII. GROUNDS FOR STANDING	78
XIII. PAYMENT OF FEES	78
XIV. STATEMENT OF PRECISE RELIEF REQUESTED	79

Patent No. 9,923,995
 Petition for *Inter Partes* Review

TABLE OF AUTHORITIES

Cases	Page(s)
<i>AC Techs., S.A. v. Amazon.com, Inc.</i> , 912 F.3d 1358 (Fed. Cir. 2019)	44
<i>Advanced Bionics v. MED-EL</i> , IPR2019-01469, Paper 6 (Feb. 13, 2020)	71
<i>Alcatel-Lucent USA v. Oyster Optics</i> , LLC, IPR2018-00070, Paper 14, 15 (May 10, 2018)	69
<i>Apple Inc. v. Fintiv, Inc.</i> , IPR2020-00019, Paper 11, 11 (Mar. 20, 2020)	73
<i>Becton, Dickinson & Co. v. B. Braun Melsungen AG</i> , IPR2017-01586, Paper 8, 17-18 (Dec. 15, 2017)	70, 71, 72
<i>CLIO USA, Inc. v. The Procter and Gamble Company</i> , IPR2013-00448, Paper 15, 3 (Feb. 4, 2014)	43
<i>Dish Network L.L.C. v. TQ Delta LLC</i> , IPR2016-01470, Paper 14, 6-7 (Feb. 9, 2017)	11
<i>Fox Factory, Inc. v. SRAM, LLC</i> , IPR2016-01876, Paper 8, 7-8 (Apr. 3, 2017)	69
<i>General Plastics. General Plastic Co., Ltd. v. Canon Kabushiki</i> <i>Kaisha</i> , IPR2016-01357, Paper 19, 16–17 (Sept. 6, 2017)	72
<i>Jenam Tech, LLC v. LG Electronics, Inc. et al.</i> , 4:19:cv-00249 (E.D. Tex.)	76
<i>Jenam Tech, LLC v. Samsung Electronics Co., Ltd. et al.</i> , 4:19:cv-00250 (E.D. Tex.)	77
<i>LG Electronics v. Advanced Micro Devices</i> , IPR2015-00329, Paper 13, 9-13 (July 10, 2015)	17

Patent No. 9,923,995

Petition for *Inter Partes* Review

TABLE OF AUTHORITIES

(continued)

	Page(s)
<i>NHK Spring v. Intri-Plex Techs.</i> , IPR2018-00752, Paper 8 (Sept. 12, 2018).....	74, 75
<i>Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.</i> , 868 F.3d 1013 (Fed. Cir. 2017)	10
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005)	10
<i>Polycom, Inc., v. Directpacket Research, Inc.</i> , IPR2019-01235, Paper 19, 28-32 (Jan. 13, 2020)	14
<i>Resideo Techs., Inc. v. Innovation Sciences, LCC</i> , IPR2019-01306, Paper 19, 11.....	74, 75
<i>Samsung Electronics Co., Ltd. v. Seven Networks, Inc.</i> , IPR2018-01106, Paper 21, 21 (Nov. 28, 2018).....	69
<i>Sud-Chemie, Inc. v. Multisorb Techs., Inc.</i> , 554 F.3d 1001 (Fed. Cir. 2009)	44
<i>Valve Corp. v. Elec. Scripting Prods., Inc.</i> , IPR2019-00062 Paper 11 (Apr. 2, 2019).....	73
<i>WAG Acquisition, LLC v. WebPower, Inc.</i> , 781 F. App'x 1007 (Fed. Cir. 2019).....	44
 Statutes, Rules and Regulations	
35 U.S.C. §102(b)	12, 15, 17
35 U.S.C. §103	<i>passim</i>
35 U.S.C. §311	1
35 U.S.C. §314(a)	74
35 U.S.C. §325(d)	69
37 C.F.R. §42.8.....	75

Patent No. 9,923,995

Petition for *Inter Partes* Review

TABLE OF AUTHORITIES
(continued)

	Page(s)
37 C.F.R. §42.8(b)(1).....	75
37 C.F.R. §42.8(b)(2).....	76
37 C.F.R. §42.8(b)(3).....	77
37 C.F.R. §42.8(b)(4).....	77
37 C.F.R. §42.15(a).....	77
37 C.F.R. §42.103	77
37 C.F.R. §42.104(a).....	77
37 C.F.R. §42.104(b)	1
37 CFR 1.102(e).....	9
83 Fed. Reg. 51340 at 51351, 51353	10

Patent No. 9,923,995

Petition for *Inter Partes* Review**PETITIONERS' EXHIBIT LIST**

No.	Reference	Referred To As
1001	U.S. Patent No. 9,923,995	'995 Patent
1002	Prosecution History of U.S. Patent No. 9,923,995	'995-FH
1003	Declaration of Scott Bradner	Bradner
1004	L. Eggert, TCP Abort Timeout Option, draft-eggert-tcm-tcp-abort-timeout-option-00, Network Working Group, Internet-Draft (April 14, 2004)	Eggert
1005	EP 1 242 882 B1 (Hankinson)	Hankinson
1006	DARPA RFC 793	RFC 793
1007	DARPA RFC 1122	RFC 1122
1008	<i>Jenam Tech, LLC v. LG Electronics, Inc. et al</i> , Case No. 4:19-cv-00249, ECF No. 1 (E.D. Tex. Apr. 3, 2019)	LG Complaint
1009	Internet Domain Survey – Jan 2010	Internet Domain Survey – Jan 2010
1010	IETF RFC 879	RFC 879
1011	IETF RFC 1072	RFC 1072
1012	IETF RFC 1323	RFC 1323
1013	IETF RFC 2018	RFC 2018
1014	IETF RFC 5482	RFC 5482
1015	IETF RFC 1131	RFC 1131
1016	IETF RFC 1771	RFC 1771

Patent No. 9,923,995

Petition for *Inter Partes* Review

1017	TCP Kind Number List- IANA	TCP Kind Number List
1018	IETF Mailing Lists	IETF Mailing Lists
1019	IETF Announce Mailing List, April 2004	IETF Announce
1020	IETF TCP Maintenance, April 2004	TCPM List
1021	Fernando Gont, TCP Adaptive User TimeOut (AUTO) Option, draft-gont-tcpm-tcp-auto-option-00.txt, IETF, Internet-Draft (May 19, 2004).	Gont
1022	Simon Schütz, Network Support for Intermittently Connected Mobile Nodes, Diploma Thesis, Department of Computer Science IV, University of Mannheim, Germany (June 13, 2004)	Schütz Thesis
1023	Cristian Zamfir, Live Migration of User Environments Across WideArea Networks, Master's Thesis, Department of Computing Science, University of Glasgow (October, 2008)	Zamfir Thesis
1024	Apogee Communications, Rapport de Veille Technologique Securite No. 69, (Security Technology Watch Report) (April, 2004)	Apogee
1025	IETF face-to-face meeting (San Diego, August 2, 2004)	tcpm-ietf60
1026	L. Eggert and F. Gont., TCP User Timeout Option, Network Working Group, Internet-Draft (May, 2005)	eggert-gont
1027	W. Richard Stevens, TCP/IP Illustrated Volume 1 The Protocols (1994)	Stevens
1028	IETF Document Management System	IETF Datatracker 1 for Eggert

Patent No. 9,923,995

Petition for *Inter Partes* Review

	Datatracker 1 for Eggert	
1029	IETF Document Management System Datatracker 2 for Eggert	IETF Datatracker 2 for Eggert
1030	Declaration of Sandy Ginoza for IETF RFC 793: (Darpa Internet Program Protocol Specification)	Ginoza RFC 793 Declaration
1031	Declaration of Alexa Morris for Eggert	Morris Eggert Declaration
1032	The Networking and Information Technology Research and Development Program, The Federal Networking Council (FNC) (Oct. 10, 1997)	FNC
1033	The Federal Networking Council (FNC), FNC Resolution: Definition of “Internet” (Oct. 24, 1995)	Internet-res
1034	United States Patent and Trademark Office, Patent Database Search Results: “transmission control protocol” (last searched Apr. 4, 2020).	Patent search-tcp-ip
1035	Google, Google Search: “tcp/ip” (last searched Apr. 4, 2020).	Google-books
1036	RFC 2026	RFC 2026
1037	RFC 2418	RFC 2418
1038	Bradner IETF documents	Bradner IETF documents
1039	Past IETF meetings	IETF meetings
1040	History of IETF tcpm working group	tcpm history
1041	Patent search for RFC 793	Patents-rfc 793
1042	Internet Archive Wayback Machine Screenshot of IETF’s website Internet- Drafts Database Interface ID-Search (Jan.,	IETF ID search page Jan. 2010

Patent No. 9,923,995

Petition for *Inter Partes* Review

	03, 2010)	
1043	Internet Archive Wayback Machine Screenshot of IETF's website IETF Tools (Jan. 01, 2010)	IETF tools page Jan. 2010
1044	Internet Archive Wayback Machine Screenshot of IETF's website IETF Browse and Search Tools (Jan. 01, 2010)	IETF tools-browse-and-search
1045	IETF, IESG Statement: Normative and Informative References (Apr. 19, 2006)	IETF RFC References
1046	U.S. Patent No. 7,404,210	US 7,404,210
1047	U.S. Patent No. 7,426,569	US 7,426,569
1048	<i>Jenam Tech, LLC v. LG Electronics, Inc. et al</i> , E.D. Tex. Case No. 4:19-cv-00249, Docket	LG Docket
1049	<i>Jenam Tech, LLC v. LG Electronics, Inc. et al</i> , E.D. Tex. Case No. 4:19-cv-00249, ECF No. 71, Revised Joint Status Report	LG Revised Joint Status Report

Patent No. 9,923,995

Petition for *Inter Partes* Review

I. INTRODUCTION

Petitioners request *inter partes* review and cancellation of claims 1-24 of U.S. Patent No. 9,923,995 (“the ’995 Patent”) (EX1001). 35 U.S.C. §311; 37 C.F.R. §42.104(b). The ’995 Patent purports to claim an invention that allows devices to negotiate a timeout period for a TCP connection. However, that was already disclosed by Eggert: “The TCP Abort Timeout Option allows conforming TCP implementations to negotiate individual, per-connection abort timeouts.” EX1004, Abstract. Accordingly, trial should be instituted.

II. TECHNOLOGY BACKGROUND

The ’995 Patent issued on March 20, 2018 and was assigned to Jenam on November 1, 2018. The ’995 Patent claims priority to U.S. Patent Application 12/714,454 (U.S. 8,219,606), filed February 27, 2010. For the purpose of this Petition only, Petitioners assume a priority date of February 27, 2010. The ’995 Patent generally relates to “sharing information for detecting an idle connection.” EX1001, Abstract.

The ’995 Patent more specifically relates to the Transmission Control Protocol (TCP) and the exchange of timeout information for a TCP connection over an internet protocol (IP) network. EX1003, ¶33. TCP is a transport layer connection-oriented protocol that had been standardized in DARPA, RFC 793, in 1981, twenty-nine years before the claimed priority date of the ’995 Patent. *Id.* ¶34. Moreover,

Patent No. 9,923,995

Petition for *Inter Partes* Review

TCP/IP had become the most prevalent network and transport layer protocols used for the Internet since 1983. *Id.* ¶¶35-39. The '995 Patent incorporates RFC 793 by reference. EX1001, 1:39-50.

As a transport layer, TCP operates in conjunction with an application layer, an internet layer, and a network access layer. EX1003, ¶¶40-44; EX1006, Fig. 2. The data from the application layer may represent user data or messages generated by applications. EX1003, ¶¶44. TCP is responsible for reliability, flow control, and correction of user data handed down by the application layer (e.g., email, web browser applications). *Id.* ¶¶43. The internet or IP layer is responsible for routing packets between networks. *Id.* ¶¶42.

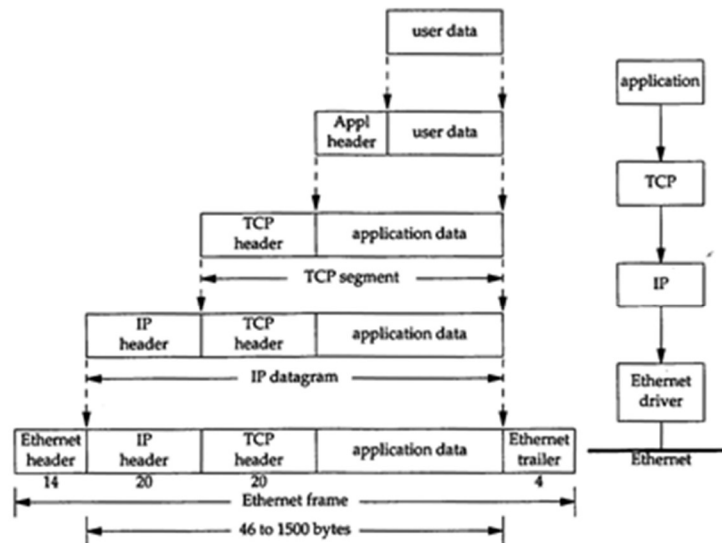


Figure 1.7 Encapsulation of data as it goes down the protocol stack.

EX1027, Fig. 1.7, 10.

Patent No. 9,923,995

Petition for *Inter Partes* Review

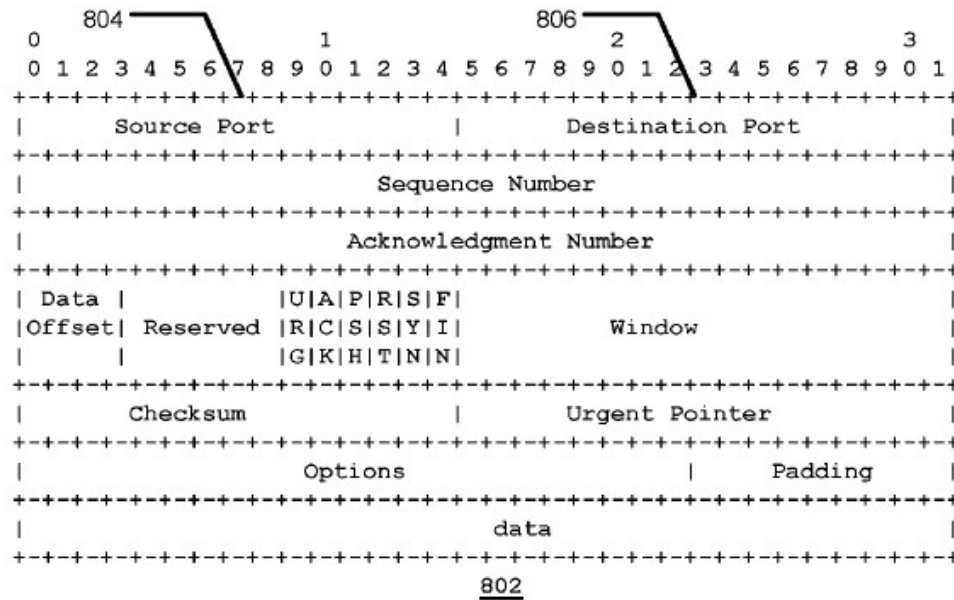
Each layer adds a header and/or a footer or trailer that contains layer-specific information to the data received from an upper layer. EX1003, ¶45-52.

The format of the TCP header is illustrated in Figure 3 of RFC 793, which is adapted in Fig. 8 of the '995 Patent. EX1006, 3.1; EX1001, Fig. 8; EX1003, ¶64-71. In the TCP header, the sequence number represents the order of the first octet of data contained in the packet. EX1006, §3.3. The SYN (synchronize) bit indicates that the sequence number in the header represents the initial sequence number for packets to be sent. EX1006, §3.1. The ACK (acknowledgment) bit indicates that the value contained in the Acknowledgment number field represents the next sequence number that the sender expects to receive in the TCP connection. *Id.*

A number of option fields of varying lengths can be incorporated at the end of the TCP header. EX1003, ¶64-66. Each TCP option contains up to three sub-fields. *Id.* A KIND sub-field is used to identify the option. *Id.*; EX1001, 14:43-47. A data subfield of one or more octets contains the actual data associated with the option, the meaning of which depends on the KIND value. *Id.* For example, for KIND=2, the data value represents the maximum segment (packet) size for the TCP connection. EX1006, §3.1. A length sub-field is used to specify the total number of octets in the option field including two octets for the KIND and LENGTH sub-fields.¹ *Id.*

¹ RFC 793 included the “Maximum Segment Size” option.

Patent No. 9,923,995

Petition for *Inter Partes* Review

TCP establishes an end-to-end connection between two devices (which can be a host or a node) by using a three-way handshake before transmitting data between the devices. EX1003, ¶56-67. This is depicted below. In a three-way handshake, **device 1**² sends a **synchronization (SYN)** request to **device 2** requesting a connection

```

+-----+-----+-----+-----+
|00000010|00000100| max seg size |
+-----+-----+-----+-----+
Kind=2 Length=4

```

EX1006, 18.

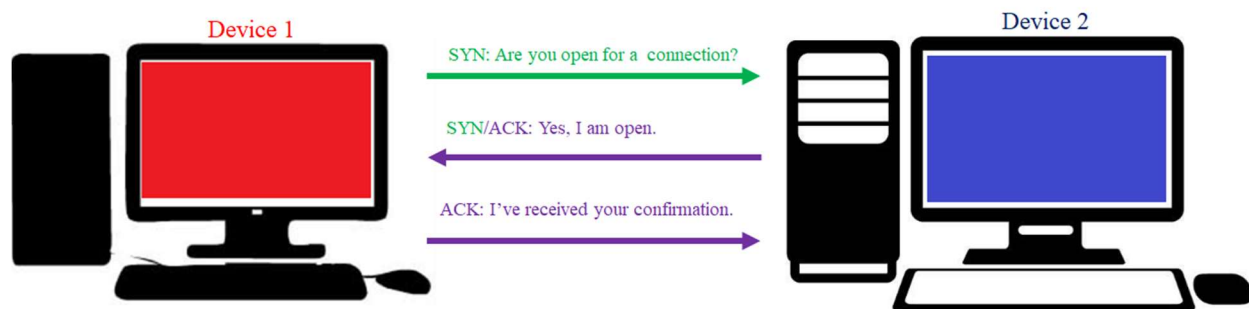
² **Red** signifies device 1 and the timeout value associated with device 1 (Timeout D1). **Blue** signifies device 2 and the timeout value associated with device 2. **Purple** is used for an acknowledgement. **Green** is used for a synchronization.

Patent No. 9,923,995

Petition for *Inter Partes* Review

and incorporating an initial sequence number for data to be transmitted by **device 1**.

Id. **Device 2** responds with a **synchronization** and **acknowledgement** (SYN/ACK) letting **device 1** know that **device 2** is open for a connection and incorporating an initial sequence number for data to be transmitted by **device 2**. *Id.* **Device 1** sends an **acknowledgement** (ACK) to **device 2** that it received its confirmation for a connection. *Id.*



TCP expects an acknowledgement for each data packet transmitted after connection establishment. EX1003, ¶59. A packet is presumed lost by the sender when an acknowledgement of the packet is not received from the destination device within a predetermined time period, called the retransmission timeout. EX1003, ¶60; EX1006, §3.7. The TCP protocol uses the acknowledgement/lack of an acknowledgement of received data to determine what data to resend to the other device. EX1003, ¶60. The sending device also uses the lack of acknowledgement from a destination as an indication that the connection between the devices is interrupted because the destination device is dead or the communication link itself is dead or disconnected. *Id.* ¶62. Thus, when no acknowledgement is received after

Patent No. 9,923,995

Petition for *Inter Partes* Review

a user timeout period, the sending device will drop the connection in order to preserve resources. EX1006, 87, 50, 77; EX1003, ¶62-63.

The original TCP protocol, however, did not specify a way to change the timeout to be used for the connection based on a value from another device. EX1003, ¶68-72.

III. THE '995 PATENT

A. Overview

The '995 Patent is directed to an apparatus for exchanging timeout information in advance of establishing a connection to specify how to modify the “timeout” period (after which the connection will be dropped) for the connection.

This is depicted in annotated Fig. 7³ of the '995 Patent below. Message 702 is received by a **first node**, for example from a local policy component, and identifies idle information (e.g., **timeout D1**) representing a duration for an idle time period (“ITP”). EX1001, 10:50-65. Message 706.1 shows a TCP packet including a header that contains ITP information (“ITP header”) sent by the **first node** and received by the **second node** to communicate the idle timeout (e.g., **timeout D1**) for the connection. EX1001, 15:26-28. The **timeout D1** in the ITP header in message 706.1

³ All color/annotations in Figures and emphasis in text was added by Petitioners.

Patent No. 9,923,995

Petition for *Inter Partes* Review

may represent a deactivation timeout to inform the **second node** how long the **first node** will keep the connection active if no data is received from the **second node**.

EX1001, 15:31-33. For example, after sending a packet 706.n, the **first node** sets a timer based on **timeout D1** (message 710.1 in Fig. 7) and deactivates the connection after the idle time period is detected by the expiration of the timer (message 712).

EX1001, 17:38-44; 17:55-57; 19:31-33.

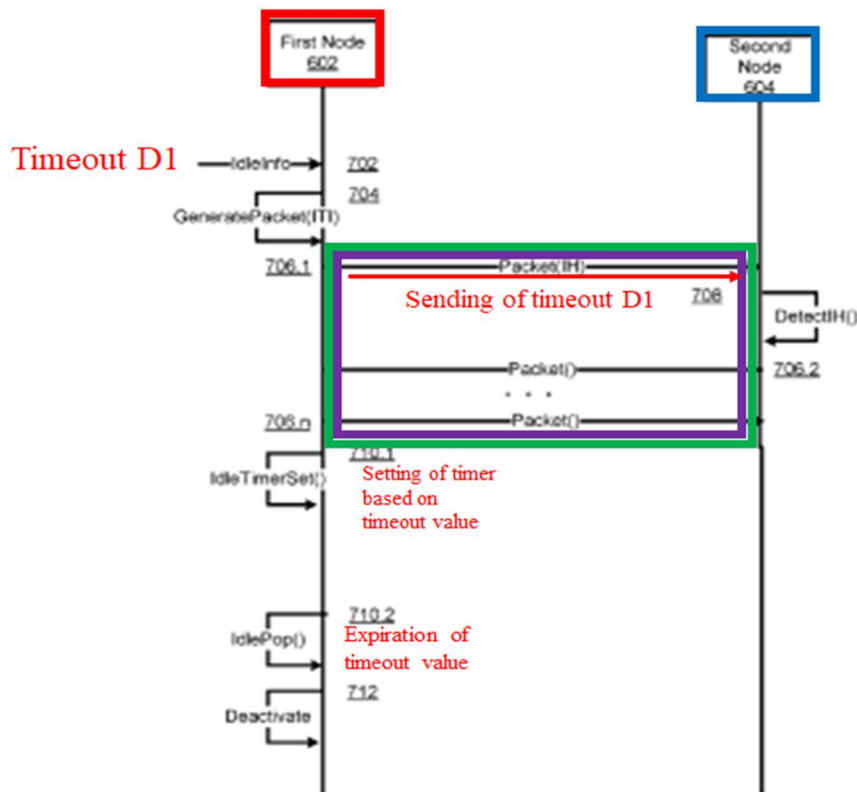


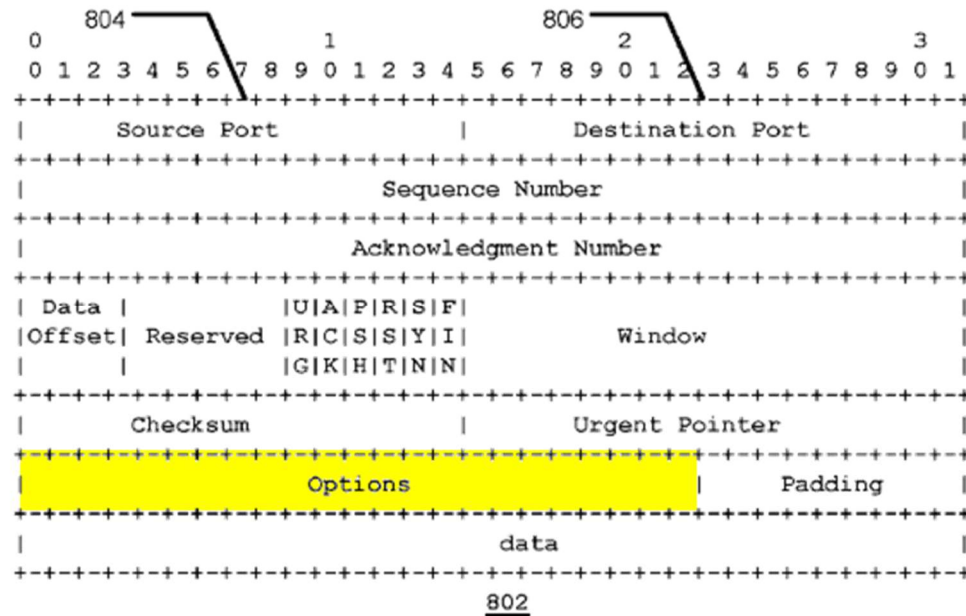
Fig. 7

The '995 Patent describes the TCP header as one possible location for transmitting the idle time period information from the **first node** to the **second node**.

Patent No. 9,923,995

Petition for *Inter Partes* Review

EX1001, Fig. 8, 14:42-50. As shown below in Fig. 8 (which patentee states is “adapted from RFC 793”), the ’995 Patent uses a TCP option field in a TCP header as a vehicle to convey the timeout value.



This option field contains three parts: a KIND octet that identifies the type of option being presented, a length sub-field that specifies the number of octets in the option field, and a third sub-field that contains data. The ’995 Patent uses the third sub-field (ITP Data) to provide the value of the timeout. EX1001, Fig. 8. Thus, the TCP option field is used as an idle time period information field to carry idle time period information. EX1003, ¶87.

Patent No. 9,923,995

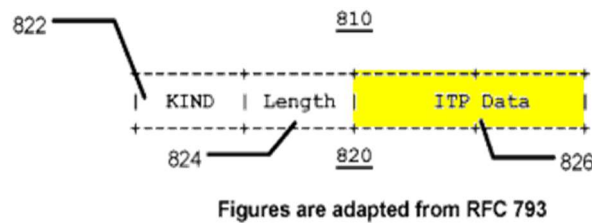
Petition for *Inter Partes* Review

Fig. 8

The idle time period can be included in the ITP header exchanged during the three-way handshake (EX1001,13:45-52) or “in a TCP packet in structures and locations other than those specified for TCP options in RFC 793.” EX1001, 14:51-57.

B. Prosecution History

The '995 Patent Application was filed on September 3, 2017, claiming priority to U.S. Patent Application 12/714,454 (U.S. 8,219,606), filed February 27, 2010. EX1002. In its initial filing, the Applicant requested prioritized examination under 37 CFR 1.102(e) and so provided no Information Disclosure Statement (IDS).

On November 29, 2017, the Examiner made his initial search, and on the same day rejected all claims only on the ground of nonstatutory obviousness-type double patenting, noting that the claims would be allowed upon receipt of a terminal disclaimer. EX1002, 218. One month later, Applicant filed a terminal disclaimer. *Id.* 237-241. At that time, some claim amendments were made. *Id.* 225-234. Finally, around the same time, Applicant filed an IDS. *Id.* 245-253.

Patent No. 9,923,995

Petition for *Inter Partes* Review

On January 23, 2018, the Examiner issued a Notice of Allowance, which included the IDS list of references cited by Applicant but without any indication that the Examiner substantially considered any reference in the IDS. *Id.* 258-264. The Applicant never cited Eggert or Hankinson to the Office.

IV. PERSON OF ORDINARY SKILL IN THE ART

For purposes of this Petition only, a person of ordinary skill in the art (POSITA) relevant to the '995 Patent would have an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking, or in the alternative, equivalent experiences, such as 6 years of work or research experience in the field of networking. EX1003, ¶¶29-32.

V. CLAIM CONSTRUCTION

Claims in *inter partes* review are construed in accordance with *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005). The Board only needs to construe terms that are subject to a legitimate dispute. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017). The PTO has recognized that constructions proposed by parties before the PTAB and the district court may differ because parties in each forum may dispute different issues. 83 Fed. Reg. 51340 at 51351, 51353.

Patent No. 9,923,995

Petition for *Inter Partes* Review

In its complaint for alleged patent infringement, Jenam defined “variant” as “manifesting variety, deviation, or disagreement; **varying usually slightly** from the standard form.” EX1008, EXC, 60. For example, the “TCP-variant” protocol disclosed in the ’995 Patent varies slightly from standard TCP by adding an idle timeout option using the format for new TCP options in a TCP header. EX1001, 14:42-57. Petitioner here applies the apparent construction that Patent Owner itself offered in District Court. *Dish Network L.L.C. v. TQ Delta LLC*, IPR2016-01470, Paper 14, 6-7 (Feb. 9, 2017), Paper 1, 11 (July 20, 2016) (accepting Patent Owner’s court construction in IPR without Petitioner acquiescing in that construction). And, as discussed below, because Eggert’s timeout option is a similar variation on TCP as disclosed in the ’995 Patent, it satisfies any construction of the term “TCP-variant.”

In addition, a number of terms in the challenged claims may be indefinite, and Petitioner may challenge those claims in the district court on that basis. For purposes of this Petition, however, Petitioners rely on the implied constructions by Jenam in the district court case.

VI. GROUNDS FORMING THE BASIS OF THIS PETITION

Ground	Claims	Statutory Basis
1	1-24	35 U.S.C. §103 (Eggert)
2	1-24	35 U.S.C. §103 (Eggert in view of Hankinson)
3	16	35 U.S.C. §103 (Eggert in view Hankinson and further in view of RFC 1122)

Patent No. 9,923,995
Petition for *Inter Partes* Review

VII. THE PRIOR ART

A. Eggert

Eggert is an IETF Internet-Draft (or “ID”) and is prior art under at least 35 U.S.C. §102(b) (pre-AIA) because it was published on April 14, 2004 more than one year before the ’995 Patent priority date.

1. Eggert is a “Printed Publication”

IDs are working documents published through the IETF Secretariat and disseminated to the public by IETF through various mediums so that others may comment on them. EX1003, ¶73-77; EX1031, 4-5. The IETF Secretariat publishes the ID on its public website (<https://ietf.org/standards/ids/>), and announcements of the publication are sent to an IETF mailing list and to relevant working group mailing lists. *Id.*; EX1031, 5-6. Anyone can subscribe to IETF mailing lists, and the archives of all IETF mailing lists are also publicly available and searchable on IETF’s website. *Id.*; EX1031, 6-8. Once published, an ID is not modified, but instead new versions are created and appended to its name, e.g.: “-00” and “-01”). EX1003, ¶78. The ID may also further mature into an RFC after passing a technical review and if the IETF management agrees. *Id.* ¶79-80, 67.

Eggert was published in April 2004. EX1003, ¶114-118; EX1031; EX1028 (Showing published document version tracking); EX1029 (Showing published document history). The declaration of Alexa Morris, Managing Director of IETF,

Patent No. 9,923,995

Petition for *Inter Partes* Review

confirms Eggert was published, disseminated, and reasonably available to the public by April 15, 2004. EX1003, ¶119-120; EX1031, 9-10. A POSITA could have learned about Eggert in various ways, such as through announcements on the IETF announce mailing list or the IETF tcpm mailing list, discussions on the IETF tcpm mailing list, review of the archives of the IETF announce or tcpm mailing lists, or attendance of a tcpm working group meeting where the document was discussed. EX1003, ¶120, 132-38; EX1031, 10.

At the time of the earliest '995 Patent priority date, it was well-known by POSITAs that IETF was home to the standards of TCP/IP—the most widely used networking protocol in the Internet at the time. EX1003, ¶122, 53-54, 67. A POSITA working on TCP-related topics, such as modifying TCP timeouts, would look first to IETF and be familiar with the tcpm working group—IETF's working group specifically charged with the task of maintaining the TCP standard. EX1003, ¶122. Therefore, such a POSITA would already be familiar with Eggert, or would have discovered it through the various tcpm announcements and discussions. EX1003, ¶122, 128-138.

In addition, a POSITA prior to 2010 could have also readily located Eggert on IETF's website through various ways, such as by browsing or searching IETF's ID Database by author, title, filename, working group, or by the use of the Google-powered search tools available on IETF's website. EX1003, ¶120. Even in the

Patent No. 9,923,995

Petition for *Inter Partes* Review

unlikely event a POSITA was unfamiliar with IETF's website, an Internet search using Google or similar search engine would lead a diligent POSITA to IETF's website or directly to Eggert prior to 2010. EX1003, ¶121, 123-126. Collectively, this demonstrates that Eggert was published and publicly accessible no later than April 2004. *Polycom, Inc., v. Directpacket Research, Inc.*, IPR2019-01235, Paper 19, 28-32 (Jan. 13, 2020) (finding a similar IETF Internet Draft was a prior art "printed publication" based on similar evidence).

2. Overview of Eggert

As in the '995 Patent, Eggert generally discloses the negotiation of a timeout value between two devices or nodes prior to establishing a connection. Eggert "allows both the **initiator** of a TCP connection (i.e., the node sending the **SYN**) as well as the **responder** of a TCP connection (i.e., the node receiving the **SYN**) to initiate an abort timeout (ATO) negotiation during the connection's three-way handshake. Figure 2 illustrates the two allowed exchanges." EX1004, 3⁴. For example, on the left side of Fig. 2, the **initiator** of a TCP connection initiates an abort timeout (ATO) negotiation by incorporating an **ATO** field as a new TCP option in the first **SYN segment** sent to the **responder**, during the three-way handshake. In this example, the **initiator** sends a **synchronization** request plus an **abort timeout option** (**SYN+ATO**). Then, **device 2, the responder**, sends a **synchronization** and

⁴ Petitioners cite to the page numbers indicated in Eggert.

Patent No. 9,923,995

Petition for *Inter Partes* Review

acknowledgment plus an abort timeout option confirming acceptance or shortening of the offer. For example: “To accept the offer, the host echoes the proposed value back to its peer by including a TCP Abort Timeout Option in its next segment. To shorten the offer, it lowers the timeout value accordingly before sending.” EX1004, 5. The initiator then responds with an acknowledgment.

On the right side of Fig. 2, the responder may also initiate the negotiation by incorporating the ATO field in the SYN/ACK segment it sends after receiving a SYN without ATO from the initiator.

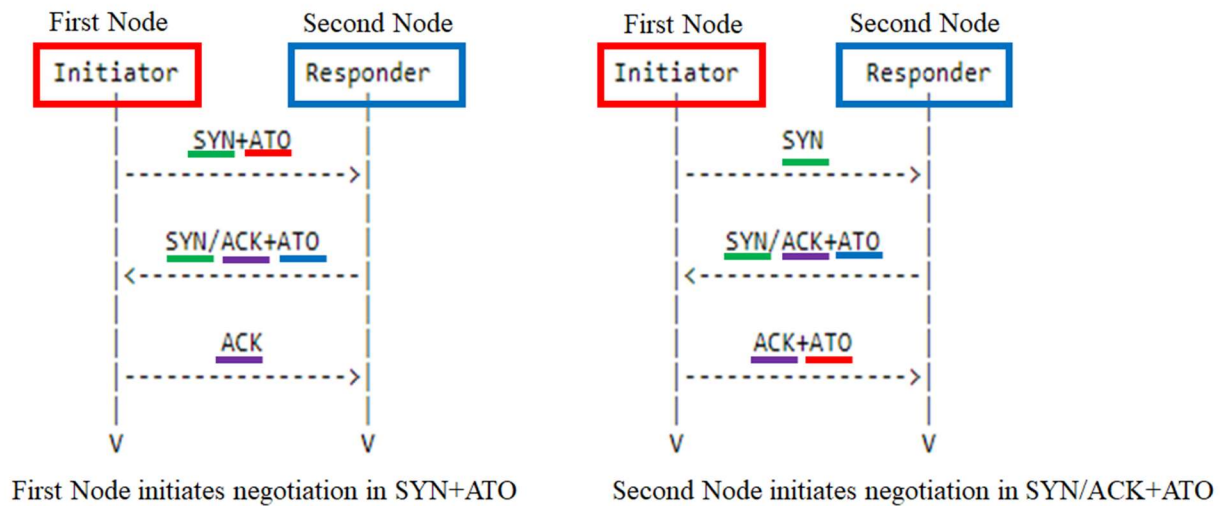


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

B. Hankinson

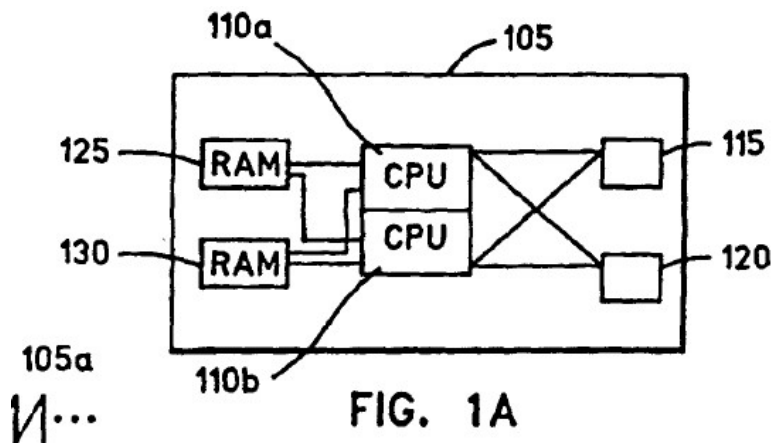
EP 1 242 882 B1 to Hankinson was published April 20, 2005 more than one year before the earliest priority date claimed by the '995 Patent and is prior art under 35 U.S.C. §102(b) (pre-AIA). Hankinson discloses a computer system that incorporates one or more CPUs to implement an operating system such as Solaris or

Patent No. 9,923,995

Petition for *Inter Partes* Review

Linux stored in a conventional hard drive, CD-ROM, or ROM. EX1005, [0033] (“at least one CPU 110a (central processing unit), . . . and a small amount of non-volatile memory (ROM) 130 to store the initial program image”).

The computer system of Hankinson also implements TCP as described in RFC 793, which is incorporated by reference in Hankinson. EX1005, [0126].



EX1005, Fig. 1.

Hankinson demonstrates that workstations or server computers such as Solaris or Linux servers were readily available with a hard disk as a non-transitory memory for storing an operating system such as Solaris, TCP protocol, and applications (e.g., HTTP server). EX1003, ¶148.

Patent No. 9,923,995

Petition for *Inter Partes* Review

C. RFC 1122

RFC 1122 was published in October 1989 more than twenty-one years before the '995 Patent earliest priority date and qualifies as prior art at least under 35 U.S.C. §102(b) (pre-AIA). EX1003, ¶153.

RFC 1122 was publicly accessible for at least two reasons. First, it is cited prior art in multiple other patents, which, in conjunction with other evidence of its publication, is “evidence of public accessibility as of a particular date.” *LG Electronics v. Advanced Micro Devices*, IPR2015-00329, Paper 13, 9-13 (July 10, 2015) (holding that an IDS together with a copyright notice is evidence of public accessibility). RFC 1122 is cited on the front page of US 7,404,210 (cited on the front of the '995 Patent) and discussed in the patent at Abstract, 2:26-34, 2:44-47, 3:30-35. EX1046. In addition, US 7,426,569 (cited in the '995 Patent) referenced RFC 1122 as other prior publication on the front page. EX1047. In addition, RFC 5482 (cited in the '995 Patent) cites RFC 1122 as a normative reference. EX1014, 5, 7-9.

Second, Petitioners' expert was involved in discussions about parts of RFC 1122 before it was published. EX1003, ¶154. After it was published, Petitioners' expert, while in the US, downloaded it multiple times from the RFC Editor ftp site (it was published before the www was invented) and a number of times from the IETF web site after the www was deployed. *Id.*

Patent No. 9,923,995

Petition for *Inter Partes* Review

VIII. CLAIMS 1-24 ARE UNPATENTABLE UNDER §103

A. Ground 1: Claims 1-24 Would Have Been Obvious Over Eggert

1. Claim 1/15

a) 1(pre)/15(pre) An apparatus comprising:

Eggert discloses an apparatus. EX1003, ¶157-63. For example, Eggert discloses modifying a TCP implementation on a computer host to support “a new TCP option - the Abort Timeout Option - that allows conforming hosts to negotiate per-connection abort timeouts.” EX1004, 2. Eggert also discloses Solaris as an example of a TCP implementation. *Id.* (“Instead of a single user timeout, some TCP implementations offer finer-grained mechanisms. For example, Solaris supports different timeouts.”); *id.* §2.1. A POSITA would have understood that Eggert refers to “Solaris” to describe a computer (Solaris computer) that incorporates the Solaris Operating System and a TCP implementation. EX1003, ¶163.

b) 1(a)/15(a) a non-transitory memory storing instructions; and

Eggert discloses that the host can be a Solaris computer. EX1003, ¶164-75. A POSITA would have found it obvious to implement Eggert on a Solaris computer including a non-transitory memory (e.g., a hard disk) to store software such as an operating system (e.g., Solaris operating system) and associated TCP protocol (e.g., Solaris’s TCP implementation) that contain instructions for the computer. EX1003, ¶164-68. A Solaris computer contains a TCP implementation and network

Patent No. 9,923,995

Petition for *Inter Partes* Review

applications (e.g., Web Server, Web browser), that incorporate software instructions.

Id. ¶168-71. It would thus be obvious to a POSITA to implement that computer with non-transitory memory to hold the operating system and the implemented communication protocol such as the TCP/IP protocol stack to be loaded when needed for execution by the host. *Id.* ¶168-75.

- c) 1(b)/15(b) one or more processors in communication with the non-transitory memory,

Eggert discloses that the host computer, which can be a Solaris computer, performs processing of the ATO. EX1004, 5; EX1003, ¶176-84. A POSITA would have found it obvious to implement Eggert's modified TCP implementation on a Solaris computer host that incorporates a processor (e.g., a CPU) to execute the instructions in the installed software such as the operating system and associated TCP protocol. EX1003, ¶176-80. A CPU conventionally communicates with non-transitory memory (e.g., the hard disk) to load software to be executed by the processor. EX1003, ¶181-84. It would have been obvious to a POSITA that a Solaris computer such as that disclosed in Eggert (EX1004, 2) would incorporate a processor such as a CPU. EX1003, ¶182-84.

- d) 1(c)/15(c) wherein the one or more processors execute the instructions for:

For the reasons in Section VIII.A.1.c, it would have been obvious to a POSITA that the processor incorporated in the host executes the modified TCP

Patent No. 9,923,995

Petition for *Inter Partes* Review

implementation of Eggert. EX1003, ¶185-89.

- e) 1(d) receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established;

Eggert discloses that a computer host operating as a responder (second node) can receive a segment (SYN+ATO) from an initiator (first node) that contains a TCP ATO in a first SYN segment⁵ transmitted during a standard TCP three-way handshake, which precedes the establishment of the standard TCP connection. EX1004, 3 (“This specification allows both the initiator of a TCP connection (i.e., the node sending the SYN) as well as the responder of a TCP connection (i.e., the node receiving the SYN) to initiate an abort timeout negotiation during the connection's three-way handshake.”); EX1003, ¶¶190-223.

Claim 1, Example 1: In Fig. 2 of Eggert below, the responder (second node)

⁵ The '995 Patent uses the term “packet” to refer to what is described in the TCP standard (RFC 793) as “segment.” EX1003, ¶88-93. Specifically, the TCP header in Fig. 8 is associated with a TCP packet, while the same header is associated with a TCP segment in RFC 793. EX1001, 14:24-26; EX1006, §3.1, 15. A POSITA would also understand that the terms “packet” in the '995 Patent and “segment” used in Eggert can be used interchangeably here. EX1003, ¶88-93. Thus, Petitioner will also use the term TCP packet to refer to a TCP segment.

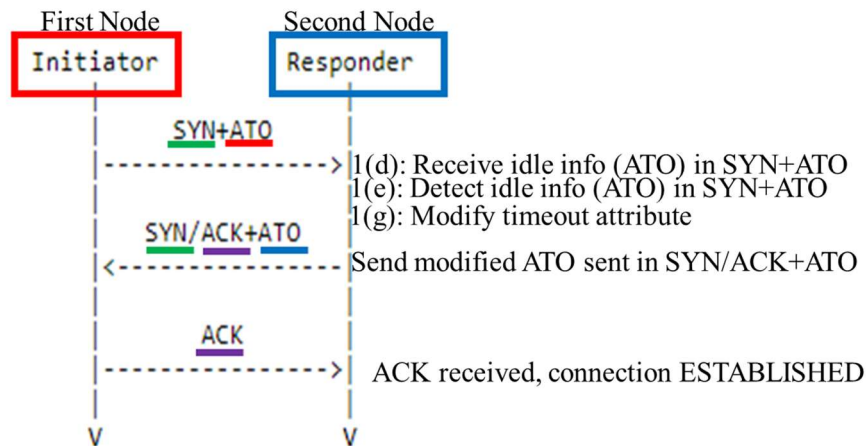
Patent No. 9,923,995

Petition for *Inter Partes* Review

may receive a SYN+ATO segment, the TCP-variant packet, containing an Abort Timeout offer from the initiator during a modified TCP three-way handshake. EX1004, 3 (“A host wishing to negotiate a specific abort timeout for a connection MAY include the TCP Abort Timeout Option in any segment with a SYN flag, i.e., either the initial SYN or the SYN-ACK.”); EX1003, ¶¶200-13. The SYN+ATO and SYN/ACK+ATO segments are the first and second SYN segments transmitted during the TCP three-way handshake. The connection is not established until after the third (ACK) segment in the handshake. *Id.* Indeed, Eggert states: “[t]he processing of the Abort Timeout Option defined in Section 2 requires timeout negotiation to occur during a connection's three-way handshake. Although this simplifies the protocol, it eliminates the possibility of negotiating new timeouts after connection establishment.” EX1004, 5. Specifically, the connection described by Eggert does not reach ESTABLISHED state until after an ACK has been exchanged between the endpoints at the end of the three-way handshake. EX1003, ¶¶208-213. Thus, the SYN+ATO and SYN/ACK+ATO must be sent in advance of the establishment of the TCP-variant connection.

Thus, the abort timeout negotiation occurs before connection establishment.

Patent No. 9,923,995

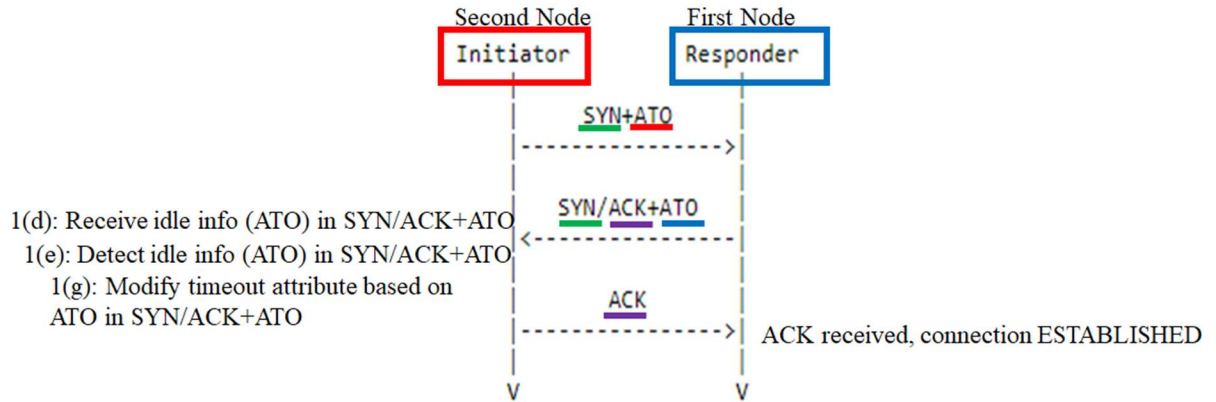
Petition for *Inter Partes* Review**Claim 1-First Example**

(Ex. 1004, Fig. 2, left side, annotated)

Claim 1, Example 2⁶: Alternatively, the **initiator (second node)** can receive a segment (**SYN/ACK+ATO**), the TCP-variant packet, from a **responder (first node)** that contains an ATO in the second step of the standard TCP three-way handshake, which is also before establishment of the connection. EX1004, 4 (“If the receiving host accepts the peer's offered abort timeout, it MUST echo the offered timeout value in the Abort Timeout Option it sends. If it shortens the timeout, it MUST send an Abort Timeout Option with a timeout value that is correspondingly less than the offered one.”); EX1003, ¶¶214-23.

⁶ Examples 1 and 2 invalidate the claims.

Patent No. 9,923,995

Petition for *Inter Partes* Review**Claim 1-Second Example**

(Ex. 1004, Fig. 2, left side, annotated)

f) 1(d) (TCP)-variant packet, TCP-variant connection;

Eggert discloses “a new TCP option - the Abort Timeout Option” to allow mobile hosts to maintain TCP connections across disconnected periods that are longer than their system's default abort timeout.” EX1004, 2; EX1003, ¶¶217-23. It explains “[a] TCP implementation that does not support the TCP Abort Timeout Option SHOULD silently ignore it.” *Id.* 4. Thus, Eggert discloses adding a new TCP option to the traditional TCP implementation. EX1003, ¶¶193-95, 214-23. Devices configured according to Eggert only vary from the “traditional TCP implementation” in the negotiation of the ATO. *Id.* ¶195, 217-220. Therefore, a connection described by Eggert using its ATO and segments sent in accordance with that option would be “TCP-variants.” *See* Section V.

Patent No. 9,923,995

Petition for *Inter Partes* Review

In addition, a modified TCP SYN segment⁷ (SYN+ATO) that contains an ATO field is a TCP-variant packet, since it uses the same format as set forth in the TCP standard but adds the ATO field. EX1003, ¶196, 451. As such, Eggert describes a segment format and corresponding protocol that “manifest[s] variety, deviation, or disagreement [the ATO option not present in RFC 793], [or] var[ies] slightly from the [TCP] standard” (by incorporating all of RFC 793), and so conforms to Jenam’s proposed construction of “variant.” *Id.* ¶196, 100; EX1008, EXC, 60.

Indeed, the Eggert ATO uses the same TCP structure as the idle time period field in the ’995 Patent Fig. 8:

Fig. 8 also illustrates a format for an exemplary ITP header 820. A KIND location is specified for including an identifier indicating that the option is an idle time period (ITP) option in an ITP header. Identifiers for option headers are currently under the control of the Internet Assigned Numbers Authority (IANA).

EX1001, 14:42-47; EX1003, ¶214-16.

⁷ The ’995 Patent uses the term “packet” to refer to “segment.” *See* footnote

Patent No. 9,923,995

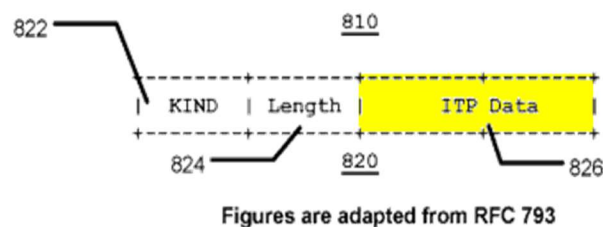
Petition for *Inter Partes* Review

Fig. 8

Similarly, Eggert's "Figure 1 shows the format of the TCP Abort Timeout Option. In Figure 1, 'X' is a TCP option number to be assigned by IANA upon publication of this document (*see* Section 5.) 'Abort Timeout' is the desired abort timeout of the connection, specified in seconds." EX1004, 2; EX1003, ¶214-16.

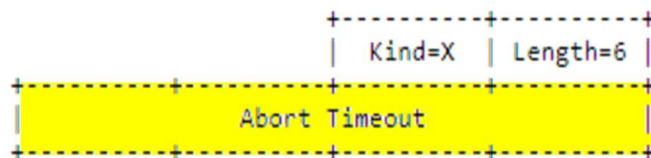


Figure 1: TCP Abort Timeout Option

As in the '995 Patent, Eggert's ATO is also incorporated in the TCP header and is formatted using: a KIND sub-field to indicate the TCP option is an ATO, a TCP option length sub-field, and the data portion of the TCP option containing the value of the ATO. EX1003, ¶216-23, 229; EX1004, 3-5. Thus, Eggert uses the same format for the ATO as the preferred embodiment disclosed in the '995 Patent for the TCP-variant packet. *Id.*

Patent No. 9,923,995

Petition for *Inter Partes* Review

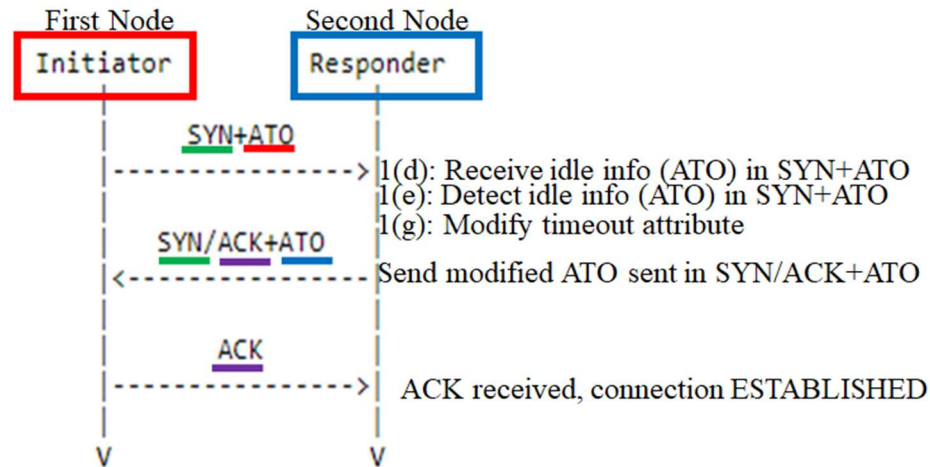
Connections established by exchanging ATOs are TCP-variant connections because they are established in accordance with a TCP-variant protocol by exchanging TCP-variant packets (SYN+ATO, SYN/ACK+ATO) during the three-way handshake. EX1003, ¶¶217-23.

- g) 1(e) detecting an idle time period parameter field in the TCP-variant packet;

Eggert discloses that the computer host operating as the responder detects an ATO field in a SYN+ATO segment. EX1004, 3 (“The timeout value included in the option specifies the proposed abort timeout for the connection. . . . Upon receipt of a segment with the Abort Timeout Option, the receiving host decides whether to accept, shorten, or reject its peer's proposed abort timeout. Section 2.3 discusses the specifics of this decision.”); EX1003, ¶¶224-32.

Claim 1, Example 1: Eggert discloses that an initiator (first node) can send a TCP segment (SYN+ATO) to a responder (second node) that contains a proposed TCP ATO in a first SYN packet (i.e., a TCP-variant packet). EX1003, ¶231.

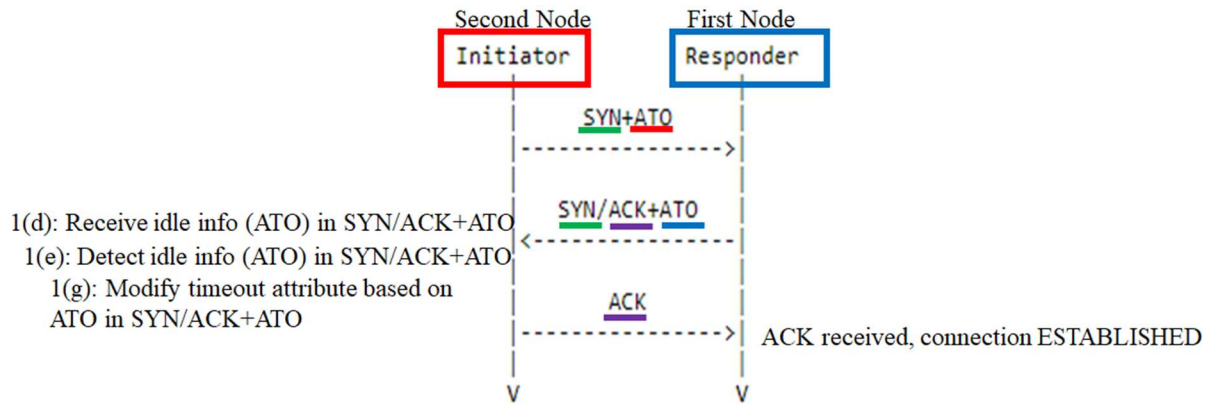
Patent No. 9,923,995

Petition for *Inter Partes* Review**Claim 1-First Example**

(Ex. 1004, Fig. 2, left side, annotated)

Claim 1, Example 2: The **initiator (second node)** detects the **ATO** field contained in the received **SYN/ACK+ATO** field from the **responder (first node)**. The **initiator** is required to use the responsive abort timeout received from the **responder**. EX1003, ¶232. To use the abort timeout, the initiator has to detect that the ATO field contains the responsive abort timeout. EX1004, 3 (“A host proposing an abort timeout to its peer **MUST** be prepared to accept a shorter timeout value than proposed after the negotiation.”).

Patent No. 9,923,995

Petition for *Inter Partes* Review**Claim 1-Second Example**

(Ex. 1004, Fig. 2, left side, annotated)

In Eggert, the ATO field is detected by using the Kind=X sub-field (which identifies the received TCP option is the new ATO) and the Length=6 field (which indicates the length of the option). EX1004, 2; EX1003, ¶¶227-30. The combination of the Kind and Length sub-fields allows the host to detect the bytes that constitute the abort timeout duration (idle time period duration or parameter) in the ATO field (idle time period parameter field). This abort timeout duration is an idle time period parameter because it specifies the time period where the connection may be idle before the connection is aborted. EX1003, ¶¶227-30; EX1004, 3-5, 9. Because it carries a proposed or responsive value for abort timeout for a connection, which represents an idle time period parameter, the ATO field is as an idle time period parameter field.

Patent No. 9,923,995

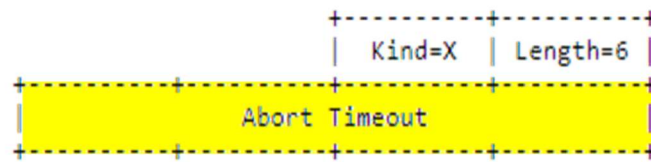
Petition for *Inter Partes* Review

Figure 1: TCP Abort Timeout Option

- h) 1(f) identifying metadata in the idle time period parameter field for an idle time period and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active

Eggert discloses “identifying metadata in the idle time period parameter field for an idle time period.” EX1003, ¶¶233-256. Eggert discloses that the **responder (second node)** processes the received **SYN+ATO** segment from the **initiator (first node)** according to an established format to identify which bits of the option represent the proposed abort timeout duration. EX1004, 2, 3, Fig. 1; EX1003, ¶¶236-42. Similarly, in the second example, the **initiator (second node)** processes the received **SYN/ACK+ATO** from the **responder (first node)** according to the established format to identify which bits of the option represent the value of the responsive abort timeout duration as accepted or shortened by the **responder**. Both the bits that identify the ATO field as being associated with an abort timeout (KIND, LENGTH) and the bits that represent the abort timeout value separately constitute metadata for the proposed or responsive abort timeout. The ’995 Patent confirms that the metadata can be a “duration of time” (EX1001, 20:56-61), which is what the

Patent No. 9,923,995

Petition for *Inter Partes* Review

ATO value contains. EX1003, ¶240; EX1004, 3-5. The abort timeout duration contained within the ATO field is an idle time period parameter because it is used to track periods during which there is no communication to keep the connection active. Because it carries an abort timeout value, which represents an idle time period parameter, the ATO field is an idle time period parameter field.

In addition, as set forth above, the format for representing the TCP abort timeout in Figure 1 of Eggert is the same as the ITP data field 826 in Fig. 8 of the '995 Patent. EX1001, 14:59-67.

Eggert also discloses that “no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active.” EX1003, ¶¶243-56. In Eggert the packet that must be “communicated . . . to keep the . . . connection active” is an ACK to a previous segment. Eggert’s ATO relies upon “[t]he TCP specification [1] [which] includes a ‘user timeout’ that defines the maximum amount of time that segments may remain unacknowledged before TCP will abort the connection. If a disconnection lasts longer than the user timeout, the TCP connection will abort.”. EX1004, 2. Like Eggert, the '995 Patent also describes modifying a TCP user timeout. EX1001, 12:44-47, 21:11-18; EX1004, 2; EX1003, ¶¶247-56. Accordingly, Eggert’s description of the abort timeout as the duration of time during which a node can wait to receive an ACK is a period “during which no packet is communicated in the TCP-variant connection to keep the TCP-variant connection

Patent No. 9,923,995
Petition for *Inter Partes* Review
active.” EX1003, ¶¶247-56.

In addition, Eggert’s ATO “allows mobile hosts to maintain TCP connections across disconnected periods that are longer than their system's default abort timeout” (EX1004, 2) and “[l]ong abort timeout values allow hosts to tolerate extended periods of disconnection” (EX1004, 5). EX1003, ¶¶244-46. This functionality parallels the ’995 Patent, which indicates that the idle time period may be associated with physical disconnection of a network medium or simply a dead connection. EX1003, ¶¶247-48; EX1001, 11:53-62, 2:17-20, 21:11-18. Using Fig. 7 below, when there is an interruption in network service as disclosed by Eggert, or a disconnected physical network medium as contemplated in the ’995 Patent (EX1001, 2:2-3; 2:17-20; 11:56-62) packets from one node do not reach the other node. In this situation, no packets are communicated to keep the connection active because they cannot get through the interruption or the disconnection in the physical medium. EX1003, ¶247-53; EX1001, 11:53-62, 2:17-20, 21:11-18.

Patent No. 9,923,995

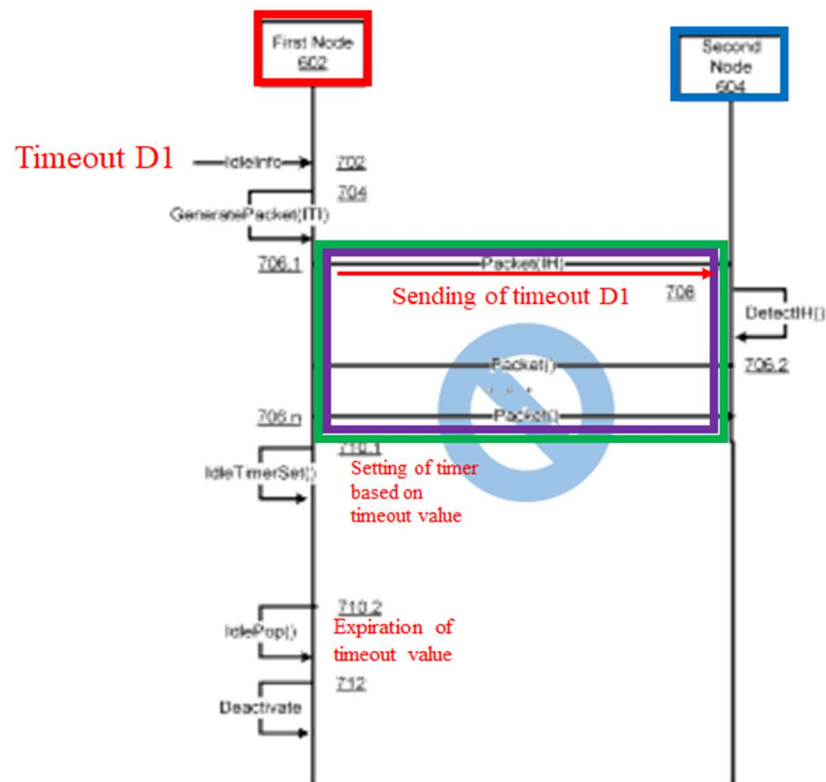
Petition for *Inter Partes* Review

Fig. 7

Thus, for both reasons, Eggert discloses an idle time period (the ATO), during which no packet is communicated in the TCP-variant connection to keep the connection active. EX1003, ¶¶247-56.

- i) 1(g) modifying, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection.

Claim 1, Example 1: In Eggert, the responder (second node) may modify the abort timeout used for the connection based on the proposed ATO received in the SYN+ATO segment from the initiator (first node). EX1004, 3; EX1003, ¶¶258-69. For example, the responder (second node) may modify the ATO from a default TCP

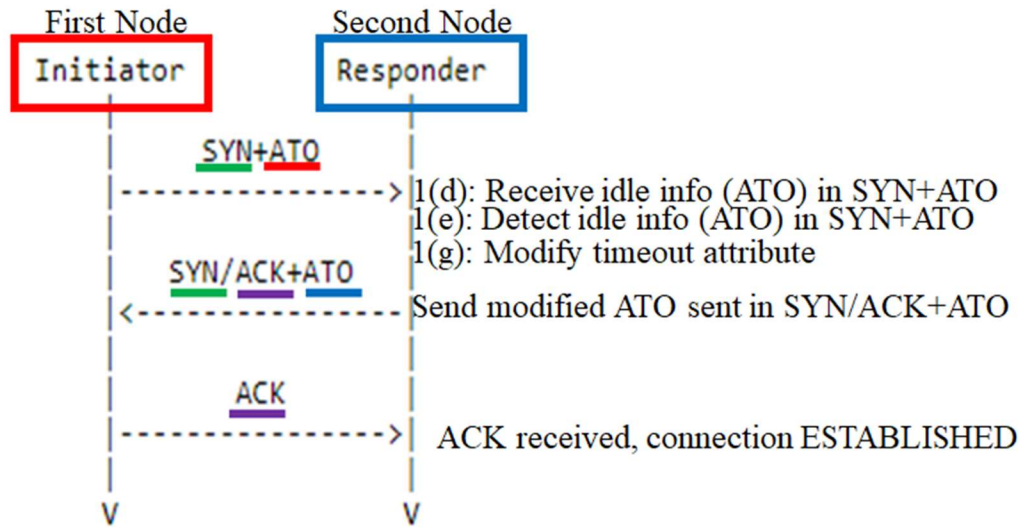
Patent No. 9,923,995

Petition for *Inter Partes* Review

user timeout to a negotiated larger value based on the proposed ATO received in the SYN+ATO segment from the initiator (first node). For instance, “[m]any TCP implementations default to user timeout values of a few minutes.” *Id.* 2. However, after successful negotiation, hosts use the abort timeout for the connection. EX1004, 4 (“If the next reply segment does not contain an Abort Timeout Option, the connection MUST use the default abort timeout. If it does, the connection MUST use the abort timeout contained inside the Abort Timeout Option.”); EX1003, ¶¶260-65, 268. As discussed above, the responder identifies the ATO and “[t]he timeout value included in the option specifies the proposed abort timeout for the connection.” EX1004, 3. The responder (second node) accepts or shortens the proposed abort timeout and uses the result as its abort timeout for the connection. EX1004, 4.

The responder (second node) in Eggert modifies the abort timeout attribute associated with the connection by using the received ATO value to, for example, change the timeout value for the connection. EX1004, 6 (“When initiators propose very long timeouts, recipients may reduce the timeout offer to an acceptable length that may still be longer than the default.”); EX1003, ¶¶265-68. Thus, the responder (second node) modifies the abort timeout associated with the connection based on the ATO (the metadata).

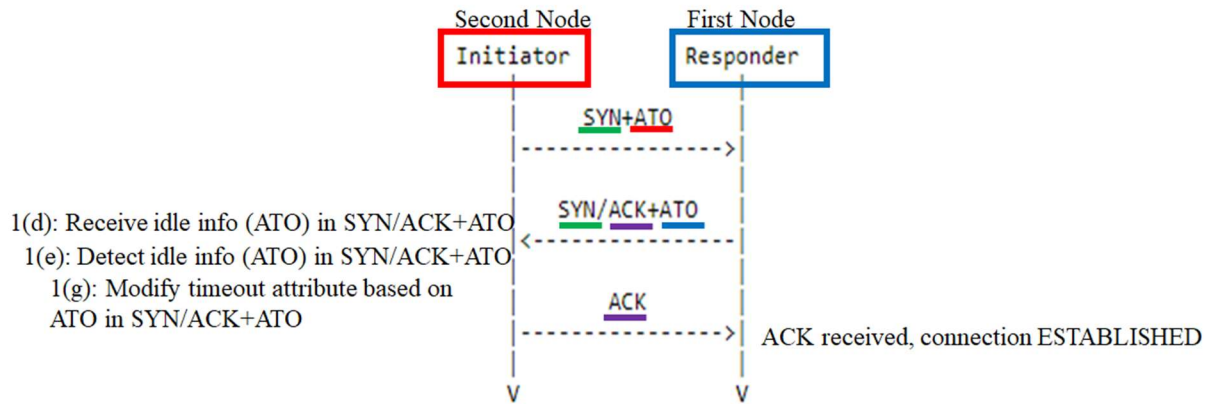
Patent No. 9,923,995

Petition for *Inter Partes* Review**Claim 1-First Example**

(Ex. 1004, Fig. 2, left side, annotated)

Claim 1, Example 2: Here, the **initiator (second node)** modifies an abort timeout attribute associated with the connection based on the **ATO** value received from the **responder (first node)** in the **SYN/ACK+ATO**. In this case, the **initiator (second node)** must use the negotiated timeout received from the **responder (first node)**. EX1004, 4; EX1003, ¶¶267-68;. Thus, the **initiator (second node)** modifies the abort timeout it uses for the connection.

Patent No. 9,923,995

Petition for *Inter Partes* Review**Claim 1-Second Example**

(Ex. 1004, Fig. 2, left side, annotated)

The '995 Patent describes timeout attributes broadly, and consistent with Eggert's abort timeout value: "ITP option handler component 562 may [*sic*] one or more attribute option handler components 564 to modify one or more corresponding attributes of a keep-alive option, a **TCP user timeout**, a retransmission timeout, an acknowledgment timeout, and another timeout associated with the TCP connection, in response to identifying the ITP header." EX1001, 21:11-18; EX1003, ¶269.

2. Claim 2

Eggert discloses "the apparatus is configured such that the timeout attribute is an attribute of a keep-alive" (claim 2). EX1003, ¶¶271-77. For example, the '995 Patent explains that "[a] receiver of a packet including an ITP header, such as second node 604, may keep a TCP connection alive based on information in the ITP header." EX1001, 15:31-33. Eggert explains that the ATO "allows mobile hosts to **maintain TCP connections** across disconnected periods." EX1004, 2. Thus, the initiator and responder in Eggert keep the TCP connection alive based on information contained

Patent No. 9,923,995

Petition for *Inter Partes* Review

in the ATO. Regarding periods of disconnection, Eggert explains that “[l]engthening abort timeouts allows established TCP connections to **survive periods of disconnection.**” EX1004, Abstract. Thus, the ATO allows the nodes to negotiate how long each node should keep the connection alive during periods of disconnection. *See* Section VIII.A.1.g incorporated by reference.

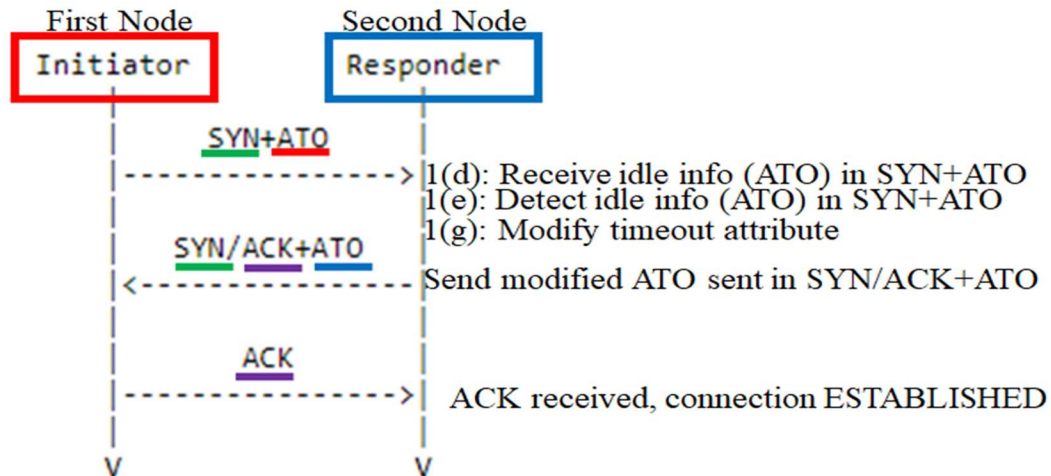
3. Claim 3

Eggert discloses at least one of Claim 3’s alternatives.

- a) 3(a) The apparatus of claim 1 wherein at least one of: the second node includes a server, the server being configured to: in response to the receiving, send, by the second node to the first node, another TCP-variant packet in advance of the TCP-variant connection being established, the another TCP-variant packet including other metadata for the idle time period;

Eggert discloses that the **receiving host (second node)** includes a **server**. EX1004, 6-7; EX1003, ¶¶279-82. The **server** can be the **responder** in Figure 2 of Eggert, and as described in Section VIII.A.1.e, responds to the **initiator (first node)** with a **SYN/ACK+ATO** segment that also contains an ATO field (containing other metadata) in advance of the TCP-variant connection being established. EX1004, 3-6; EX1003, ¶¶283-91.

Patent No. 9,923,995

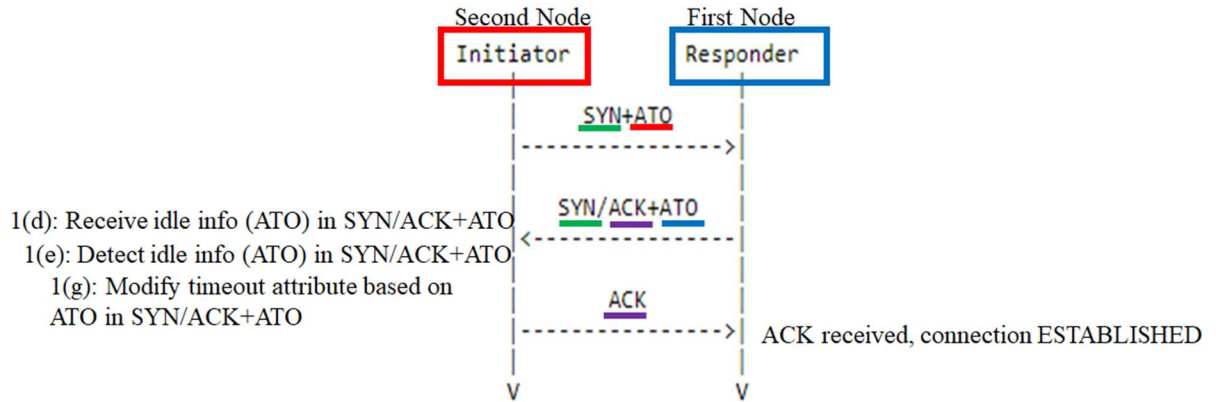
Petition for *Inter Partes* Review**Claim 1-First Example**

(Ex. 1004, Fig. 2, left side, annotated)

- b) 3(b) the second node includes a client, the client being configured such that the receiving is performed subsequent to sending, by the second node to the first node, another TCP-variant packet in advance of the TCP-variant connection being established including other metadata for the idle time period.

Alternatively, Eggert discloses that the **initiating host** can be a **client (second node)**. EX1004, 1-2; EX1003, ¶¶292-98. The **initiator (client)** in Figure 2 can receive **SYN/ACK+ATO** from the **responder (first node)** subsequent to sending the **SYN+ATO** to the **responder**, both of which occur during the three-way handshake, before the TCP-variant connection is established. EX1004, 2-3; See Section VIII.A.1.e (TCP-variant nature of the SYN/ATO segment); EX1003, ¶¶292-98, 192, 212, 221.

Patent No. 9,923,995

Petition for *Inter Partes* Review**Claim 1-Second Example**

(Ex. 1004, Fig. 2, left side, annotated)

4. Claims 4 and 5

Eggert discloses “regardless as to whether the apparatus is the server or the client, the metadata is the same as the other metadata” (claim 4) and “regardless as to whether the apparatus is the server or the client, the metadata is different from the other metadata” (claim 5). EX1003, ¶¶299-316.

For claim 4, the receiving host may accept and echo the received abort timeout value, thus the metadata and the other metadata will be the same. EX1004, 3-4, 6; EX1003, ¶¶299-306.

For Claim 5, Eggert discloses the use of different metadata in the scenario where the abort timeout is shortened: “If the receiving host . . . shortens the timeout, it MUST send an Abort Timeout Option with a timeout value that is correspondingly less than the offered one.” EX1004, 3-6; EX1003, ¶¶307-16.

In Eggert, the initiator and responder can both be either a server or client.

Patent No. 9,923,995

Petition for *Inter Partes* Review

EX1004, 3; EX1003, ¶¶312-13. The functionality described in Eggert does not change. EX1003, ¶313.

5. Claims 6/19

Eggert discloses “the timeout attribute is specified in at least one of a number of seconds or minutes” (claim 6) and “the apparatus is configured such that the timeout attribute is specified in a number of seconds” (claim 19). Eggert discloses that “the Abort Timeout Option specifies abort timeouts as 32-bit values with a granularity of **seconds** (Section 2).” EX1004, 6; EX1003, ¶¶317-22.

6. Claim 7

Eggert discloses “the timeout attribute is used to keep the TCP-variant connection open when inactive, and to prevent one or more other nodes from closing the TCP-variant connection when inactive” (claim 7). EX1003, ¶¶323-32. The negotiated abort timeout attribute specifies how long the two nodes agree “to maintain TCP connections across disconnected periods that are longer than their system’s default abort timeout.” EX1004, 2; EX1003, ¶¶327-28. It does so by keeping alive or maintaining an inactive connection open while waiting for data to be acknowledged. EX1004, Abstract, 2, 5; EX1003, ¶329-32. Normally, “[i]f a disconnection lasts longer than the user timeout, the TCP connection will abort.” EX1004, 2. The negotiation of a timeout larger than the default user timeout, per Eggert, prevents a node from closing the inactive connection before the expiration

Patent No. 9,923,995

Petition for *Inter Partes* Review

of the negotiated abort timeout. EX1004, 6 (“When initiators propose very long timeouts, recipients may reduce the timeout offer to an acceptable length that may still be longer than the default.”).

7. Claim 8

Eggert discloses “the apparatus is configured such that the metadata is used as input of an algorithm for determining a duration of time specified by the timeout attribute” (claim 8). EX1003, ¶¶333-44. In Eggert, the receiving host performs an algorithm to decide whether to shorten, accept, or reject the proposed ATO. EX1004, 3, 5. For example, the receiving host may shorten the proposed timeout to an acceptable length that is lower than the proposed timeout while being longer than a default value. EX1004, 6. The algorithm for determining whether to accept or shorten the proposed ATO includes checking with a local policy. *Id.* 5 (“The decision of whether to accept, shorten, or reject an offered timeout is a **local policy decision**. One disclosed algorithm in Eggert is to “require **prior peer authentication**, . . . before accepting long abort timeouts for the peer's connections.”); *Id.* 7. Another algorithm “would introduce a **per-peer limit** on the number of connections that may use increased abort timeouts.” *Id.* A POSITA would have found these algorithms obvious to implement. EX1003, ¶¶338-42. This is consistent with the ’995 Patent in which a policy component (e.g., ITP Policy 450) uses as an algorithm a policy or a rule for specifying the idle time period. EX1001, 11:37-52, 16:60-63.

Patent No. 9,923,995

Petition for *Inter Partes* Review

8. Claim 9

Eggert discloses “the apparatus is configured such that the algorithm is determined based on at least one particular attribute” (claim 9). EX1003, ¶¶345-54. In Eggert, the algorithm may be determined based on attributes, including a security feature or a confirmation of prior peer authentication. EX1004, 7. Or the algorithm may be based on the number of connections to limit the use of increased abort timeouts. *Id.* In addition, it would be obvious to a POSITA to implement a system where one of the policy decisions would be to choose which algorithm is most suitable for the connection. *Id.* 5; EX1003, ¶354. This is consistent with the specification of the ’995 Patent, which provides as examples security attributes and availability of a resource. EX1001, 11:53-62.

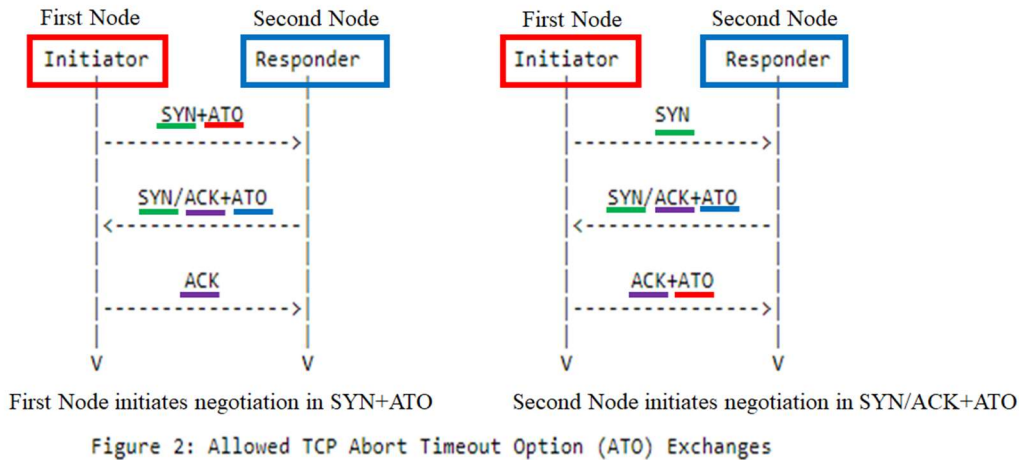
9. Claims 10/20

Eggert discloses “the apparatus is configured such that the modification of the timeout attribute results from a negotiation between the first node and the second node via a negotiation protocol of a TCP-variant protocol” (claim 10) and “the apparatus is configured such that the modification of the timeout attribute results from a negotiation between the first node and the second node” (claim 20). EX1003, ¶¶355-62, 500-04. The Eggert abort timeout value is negotiated during the connection’s three-way handshake. EX1004, 3, 5; EX1003, ¶360. The exchange of segments that include TCP ATOs between the two nodes forming a connection

Patent No. 9,923,995

Petition for *Inter Partes* Review

constitute a negotiation of the timeout attribute and therefore a negotiation protocol of a TCP-variant protocol. EX1003, ¶¶360-62; *see* Section VIII.A.1.e (TCP-variant protocol connection).



10. Claim 11/21

a) [11.a]/[21.a]

Eggert discloses “the one or more processors execute the instructions for: detecting the idle time period based on the timeout attribute” (claim 11) and “based on the idle information” (claim 21). EX1003, ¶¶363-82, 505-14. For example, Eggert discloses that the host will use the negotiated abort timeout, rather than the default user timeout, to detect whether the idle time period, i.e., how long segments remain unacknowledged, has expired. EX1004, 2 (“The TCP specification [1] includes a ‘user timeout’ that defines the maximum amount of time that segments may remain unacknowledged before TCP will abort the connection.”). Moreover, Eggert explains that the “user timeout” is determined by idle information such as the

Patent No. 9,923,995

Petition for *Inter Partes* Review

default user timeout or the ATO. For instance, “[m]any TCP implementations default to user timeout values of a few minutes.” EX1004, 2; EX1003, ¶¶368-69. However, after successful negotiation, hosts use the abort timeout for the connection. EX1004, 4; EX1003, ¶¶368-69, 509-12. “If the next reply segment does not contain an Abort Timeout Option, the connection MUST use the default abort timeout. If it does, the connection MUST use the abort timeout contained inside the Abort Timeout Option.” EX1004, 3. A POSITA would find it obvious to implement Eggert with a node that would execute instructions to measure the amount of time that segments remain unacknowledged and detect when the time elapsed reaches or exceeds the negotiated abort timeout. EX1003, ¶¶510-12.

b) [11.b]/[21.b]

Eggert discloses “in response to detecting the idle time period, deactivating the TCP-variant connection by releasing a resource allocated for the TCP-variant connection by one of the first and second nodes without signaling another one of the first and second nodes that is related to the detection of the idle time period” (claims 11/21). EX1003, ¶¶370-79, 513-14. Eggert discloses that “[i]f a disconnection lasts longer than the user timeout, the TCP connection will abort.” EX1004, 2. By aborting a connection, the host releases a resource associated with maintaining state information for the connection. The ATO of Eggert does not specify any mechanism to signal another node that an idle time period has been detected. “Negative

Patent No. 9,923,995

Petition for *Inter Partes* Review

limitations may be satisfied by silence in the prior art.” *CLIO USA, Inc. v. The Procter and Gamble Company*, IPR2013-00448, Paper 15, 3 (Feb. 4, 2014). “[A] reference need not state a feature's absence in order to disclose a negative limitation.” *AC Techs., S.A. v. Amazon.com, Inc.*, 912 F.3d 1358, 1367 (Fed. Cir. 2019) (citing *Sud-Chemie, Inc. v. Multisorb Techs., Inc.*, 554 F.3d 1001, 1004-05 (Fed. Cir. 2009)); *WAG Acquisition, LLC v. WebPower, Inc.*, 781 F. App'x 1007, 1013 (Fed. Cir. 2019) (finding that while the reference “does not specify that a pointer is not used, nothing in the record suggests that a pointer must be used” in finding substantial evidence supported the Board’s finding that the reference disclosed “the negative limitation”). A POSITA would understand that Eggert does not require the signaling of the other node if a timeout has been detected. EX1003, ¶¶375-77; EX1006, 32-34, 36. Further, a POSITA implementing Eggert would not signal the other node. *Id.* Indeed, the Eggert system assumes a disconnection occurred so the signal may not reach the other node.

Patent Owner may argue that TCP as modified by Eggert allows for the Responding Host to abort the connection by sending a specific RESET message to the remote host. EX1006, 50. However, the RESET message is not sent when the user timeout expires. EX1006, 77 (“For any state if the user timeout expires, flush all queues, signal the user "error: connection aborted due to user timeout" in general and for any outstanding calls, delete the TCB, enter the CLOSED state and return.”).

Patent No. 9,923,995

Petition for *Inter Partes* Review

Thus, on user timeout (or abort timeout), the aborted connection is closed without sending a RESET message to the remote host.” EX1003, ¶379. Moreover, as acknowledged by the ’995 Patent, the RESET message does not contain a signal related to the detection of the idle time period. EX1001, 19:34-46. Thus, Eggert does not specify any mechanism to signal the other node that an idle time period has been detected. EX1003, ¶¶374-80.

11. Claim 12/22

a) [12.a]

Eggert discloses “the apparatus is configured such that at least one of the detecting or the identifying is performed at a TCP-variant layer other than a TCP layer” (claim 12). EX1003, ¶¶381-88, 516-521. Specifically, Eggert proposes a variation to the TCP protocol by incorporating the ATO field in TCP SYN (e.g., SYN+ATO) and ACK (e.g., SYN/ACK+ATO) segments exchanged between the initiator and the responder. EX1004, 3; EX1003, ¶¶385-87. Because the TCP-variant protocol of Eggert is different from the standard TCP (described in RFC 793), a POSITA would have understood that the modified TCP of Eggert constitutes a transport layer that is a TCP-variant layer that is different from the TCP layer set forth in RFC 793. EX1003, ¶387, 520-21.

b) [12.b]

Eggert discloses “where the TCP-variant layer is above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer” (claim

Patent No. 9,923,995

Petition for *Inter Partes* Review

12) and “that at least the generating is performed at a TCP-variant layer other than a TCP layer, where the TCP-variant layer is above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer” (claim 22). EX1003, ¶¶388-92, 522-23.

HTTP applications are application layer protocols that run above transport layer protocols (e.g., TCP), which run above the network layer (e.g., IP layer). EX1003, ¶¶390-92, 522-23. Eggert modifies the TCP protocol. Thus, the obvious way for a POSITA to implement Eggert would be to have the modified TCP protocol between the application layer (e.g., HTTP) and the IP layer, just like the original TCP layer. *Id.*

12. Claims 13/23

a) [13.a]/[23.a]

Eggert discloses “the one or more processors execute a network application that is configured to perform a 3-way TCP handshake for establishing a TCP connection that is different than the TCP-variant connection” (claim 13) and the “one or more processors execute a network application that is configured to perform, in another scenario, a 3-way TCP handshake for establishing a TCP connection that is different than the TCP-variant connection” (claim 23). EX1003, ¶¶396-403, 527-28. Eggert discloses a host incorporating a modified TCP implementation software as a network application that can operate using TCP or using the TCP-variant protocol

Patent No. 9,923,995

Petition for *Inter Partes* Review

disclosed by Eggert. EX1004, 5-7. The TCP-variant protocol disclosed by Eggert is an optional one. *Id.* 7.

When supporting TCP, Eggert's software running in the processors in the server would start up communication connections using the TCP-3-way handshake without incorporating the optional ATO described above in relation to claim 1(d). The Eggert timeout negotiation would only take place with the hosts that supported and initiated the Eggert ATO. EX1003, ¶399. But "TCP implementations that support the TCP Abort Timeout Option SHOULD ignore it when they receive the option in a segment other than a SYN or first ACK." EX1004, 4. A host that supports the ATO may perform TCP connection's three-way handshake without invoking the ATO, thereby establishing a standard TCP connection with the other host. EX1003, ¶¶400-01. That standard TCP connection would be established with a three-way TCP handshake, without invoking the ATO. *Id.*

A POSITA would have found it obvious to implement Eggert with a host, for example (e.g., a server), that supports both the standard TCP and Eggert's TCP modification. *Id.* ¶402. This host would use TCP to communicate with the hosts that did not support the Eggert TCP-variant protocol and use the Eggert TCP-variant protocol only with the hosts that supported the option. *Id.*; EX1004, 4 ("A TCP implementation that does not support the TCP Abort Timeout."). Thus, such a server that supports the Eggert TCP-variant protocol would also support standard TCP

Patent No. 9,923,995
 Petition for *Inter Partes* Review
 protocol. *Id.*

b) [13.b]

Eggert discloses “wherein the network application is configured to establish the TCP-variant connection instead of the TCP connection in order to permit negotiation, between the first node and the second node, of the timeout attribute” (claim 13). EX1003, ¶¶404-09. For example, Eggert discloses that the standard TCP specification does not provide a way to negotiate the value of the “user timeout,” but conventional TCP implementations use locally defined default values for the user timeout. EX1004, 2. Thus, “instead of [using] the TCP connection,” a host that wishes to negotiate the timeout must resort to exchange Eggert’s ATO parameters during the three-way handshake. *Id.* 3 (“A host wishing to negotiate a specific abort timeout for a connection MAY include the TCP Abort Timeout Option in any segment with a SYN flag, i.e., either the initial SYN or the SYN-ACK.”). The host incorporating Eggert’s modified TCP software as a network application would be configured to use the Eggert TCP-variant protocol to permit negotiating a timeout value with hosts that support the ATO, and to use the traditional TCP protocol with hosts that do not support the ATO.

c) [23.b]

Eggert discloses “the network application is configured to establish the TCP-variant connection instead of the TCP connection and thus permit a negotiation,

Patent No. 9,923,995

Petition for *Inter Partes* Review

between the first node and the second node, of the timeout attribute” (claim 23).

EX1003, ¶¶529-31. For example, Eggert discloses that the standard TCP specification does not provide a way to negotiate the value of the “user timeout.”

Thus, a host that wishes to negotiate the timeout must exchange ATO parameters during the three-way handshake. EX1004, 3, 5. Thus, the operating system (e.g., Solaris) that incorporates a TCP implementation that supports ATO is a network application that permits negotiation of the timeout attribute.

d) [13.c]/[23.c]

Eggert discloses “where the timeout attribute is not available when establishing the TCP connection, but is available when establishing the TCP-variant connection” (claims 13/23). EX1003, ¶¶410-15, 532-37. The negotiated abort timeout attribute of Eggert is not available when establishing the TCP connection but is available when establishing the Eggert TCP-variant connection. EX1001, 2:17-20; EX1004, 5-7.

Eggert discloses that the standard TCP specification does not provide a way to negotiate the value of the “user timeout.” EX1004, 3. Thus, a host that wishes to negotiate the timeout has to resort to exchanging ATO parameters during the three-way handshake. *Id.* 5. Specifically, the ATO is new to the TCP specification and is not in the original specification. EX1003, ¶413, 535. For instance, “[m]any TCP implementations default to user timeout values of a few minutes.” EX1004, 2.

Patent No. 9,923,995

Petition for *Inter Partes* Review

However, after successful negotiation, hosts use the abort timeout for the connection.

Id. 4 (“If the next reply segment does not contain an Abort Timeout Option, the connection MUST use the default abort timeout. If it does, the connection MUST use the abort timeout contained inside the Abort Timeout Option.”). Specifically, prior to Eggert, RFC 793 provided only three options with KIND=0 for end of option list, KIND=1 for no-operation, and KIND=2 for maximum segment size. EX1006, 18.

Currently defined options include (kind indicated in octal):

Kind	Length	Meaning
----	-----	-----
0	-	End of option list.
1	-	No-Operation.
2	4	Maximum Segment Size.

e) [13.d]/[23.d]

Eggert discloses “that the TCP-variant connection is capable of being at least partially closed when inactive based on the timeout attribute” (claims 13/23). EX1003, ¶417-18, 539.

As described in Section VIII.A.1.f, Eggert’s TCP-variant connection is fully closed, meeting the limitation of being “at least partially closed,” when the host determines that the connection is inactive based on the abort timeout. EX1004, 2.

13. Claim 14

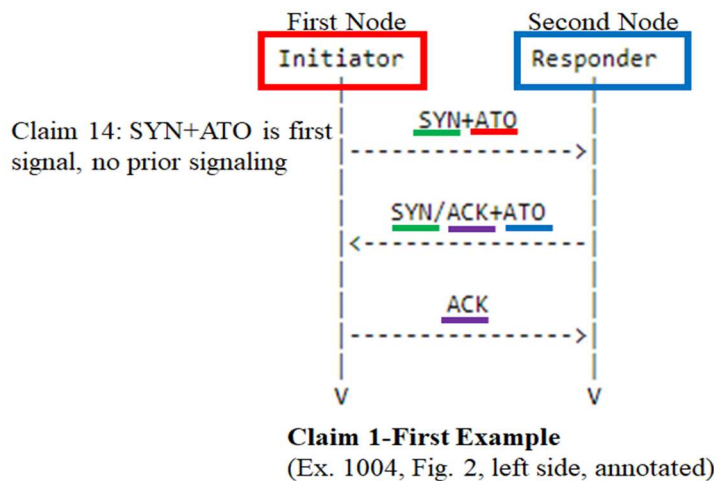
Eggert discloses “the apparatus is configured such that the TCP-variant packet and the metadata included therewith are received by the second node from the first

Patent No. 9,923,995

Petition for *Inter Partes* Review

node, without any prior signaling from the second node to the first node” (claim 14).

EX1003, ¶423-25. The messages exchanged in the Eggert three-way handshake are the first messages received by any node to establish the Eggert TCP-variant connection. As shown in Fig. 2, the SYN+ATO segment (which is the TCP-variant packet and includes the claimed metadata as described in Claims 1 and 3 is the first message transmitted for establishing a connection between the **initiator (first node)** and the **responder (second node)**. Thus, the **ATO** field is received by the **second node (responder)** from the **first node (initiator)**, without any prior signaling from the **second node** to the **first node**.



14. Claim 15

For elements 15(pre)-15(c) see Sections VII.A.1.a.-d incorporated by reference.

- a) 15(d) receiving idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to

Patent No. 9,923,995

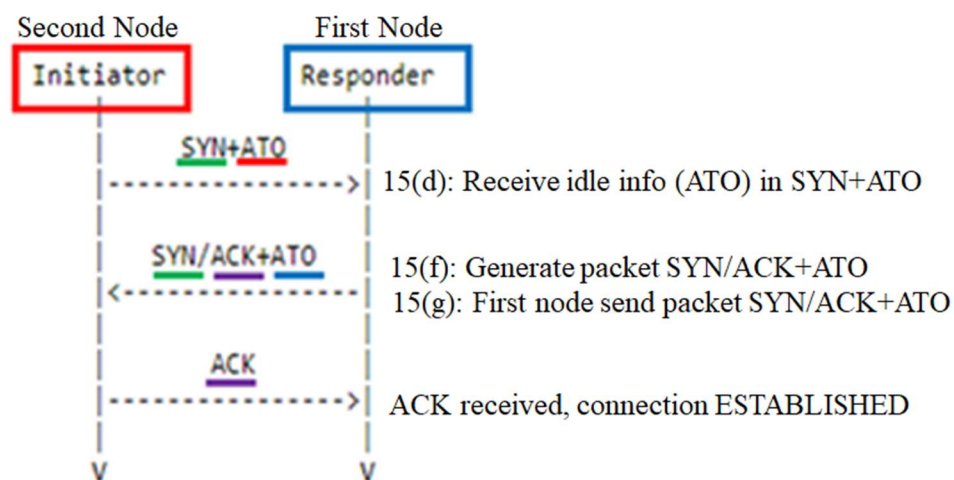
Petition for *Inter Partes* Review

keep the TCP-variant connection active:

Eggert discloses “receiving idle information for detecting an idle time period.”

EX1003, ¶¶435-43.

Claim 15, Example 1: Eggert discloses that a computer host operating as a **responder** can receive a segment (**SYN**+**ATO**) that contains a TCP **ATO** in a first **SYN** packet transmitted during a TCP three-way handshake. EX1004, 3, 5, Fig. 2 (left side). Eggert discloses that the **responder** detects an **ATO** incorporated in the received **SYN**+**ATO** segment. *Id.*



Claim 15- First Example
(Ex. 1004, Fig. 2, left side, annotated)

The **SYN**+**ATO** segment represents or includes idle information because it contains an abort timeout sub-field. The value in the abort timeout sub-field is used to determine how much time needs to pass before the timeout occurs, and therefore contains “idle information” that is used to detect the idle time period during which

Patent No. 9,923,995

Petition for *Inter Partes* Review

no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active.

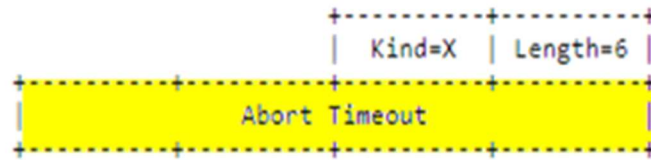


Figure 1: TCP Abort Timeout Option

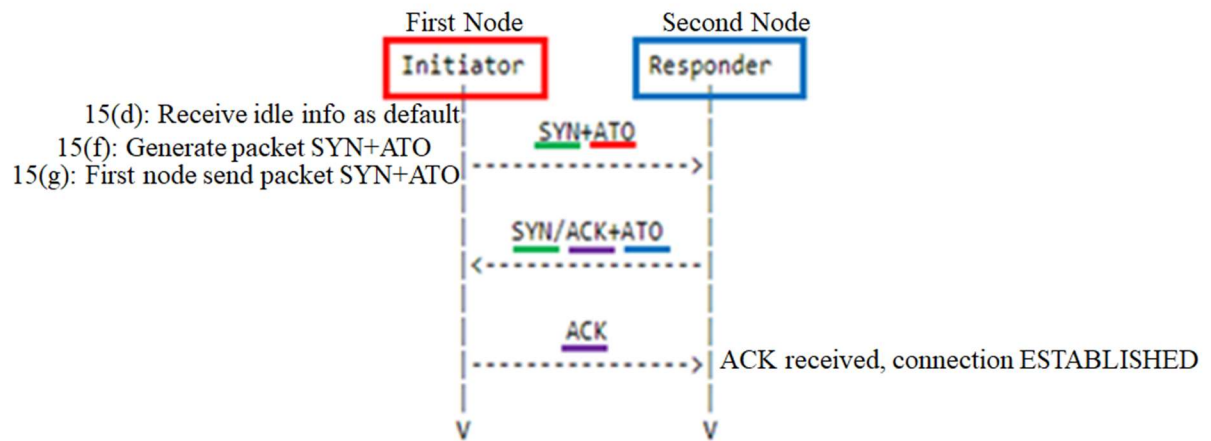
Claim 15, Example 2⁸: Eggert discloses that the **initiator** may receive idle information in the form of a default timeout or system's default abort timeout. EX1004, 2 ("Many TCP implementations default to user timeout values of a few minutes [7]."); *Id.* 4 ("The host that initially proposed the Abort Timeout Option analyzes the next segment it receives from its peer. If the next reply segment does not contain an Abort Timeout Option, the connection **MUST** use **the default abort timeout.**"). Similar to the received abort timeout sub-field, this default timeout could be used by the initiator to determine how much time needs to pass before a timeout occurs. According to the '995 Patent, idle information may be received from the memory of the node, from a settings service at the node, from a user, or from an administrator. EX1001, 11:14-19 ("received from a user and/or administrator"), 10:59-65 ("configuration storage location"). In addition, a default abort timeout is

⁸ Examples 1 and 2 invalidate the claims.

Patent No. 9,923,995

Petition for *Inter Partes* Review

a configuration of the TCP implementation received from a user or administrator.



Claim 15- Second Example

(Ex. 1004, Fig. 2, left side, annotated)

As discussed in Section VIII.A.1.e.(1), connections established by exchanging SYN+ATO and SYN/ACK+ATO segments are TCP-variant connections. As also discussed in Section VIII.A.1, during the idle time period of Eggert, no packet is communicated to keep the connection active.

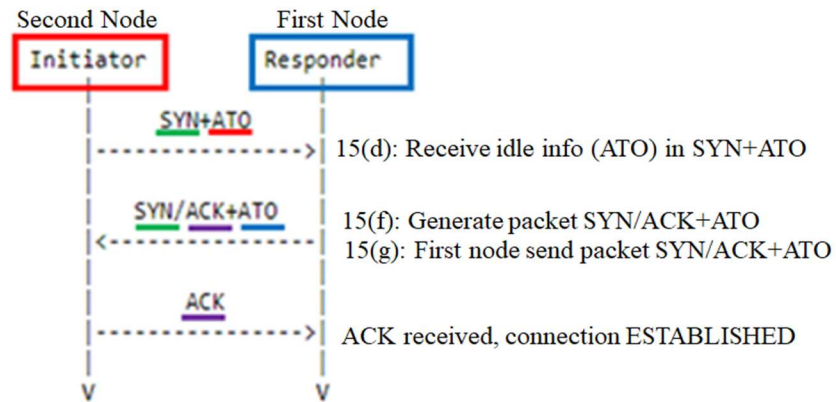
- b) 15(e) generating a TCP-variant packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information; and

Claim 15, Example 1: Eggert discloses that the computer host operating as the responder generates a SYN/ACK+ATO segment that contains a TCP ATO according to an established format in which bits of the option represent the value of a negotiated timeout to be used for a connection. EX1004, 2, 5 (left side of Fig. 2); EX1003, ¶¶444-45. The negotiated timeout value embedded in the SYN/ACK+ATO

Patent No. 9,923,995

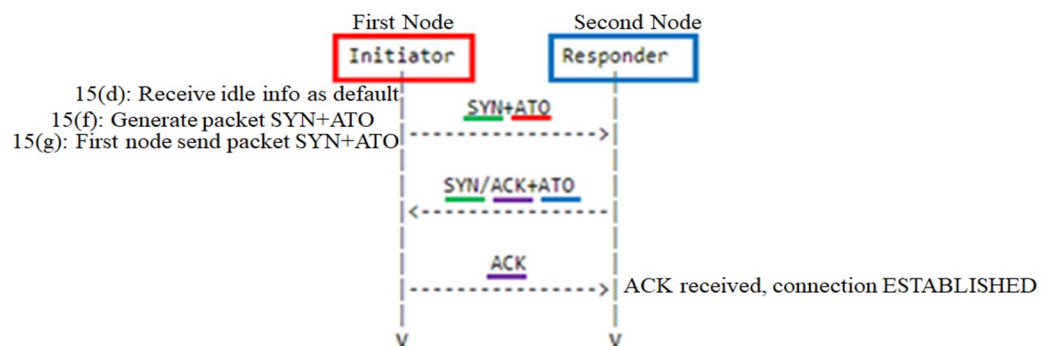
Petition for *Inter Partes* Review

by the responder may be an echo or a modified version of the timeout received from the initiator. EX1004, 3.



Claim 15- First Example
(Ex. 1004, Fig. 2, left side, annotated)

Claim 15, Example 2: The initiator generates a SYN+ATO segment that contains the TCP ATO.



Claim 15- Second Example
(Ex. 1004, Fig. 2, left side, annotated)

Patent No. 9,923,995

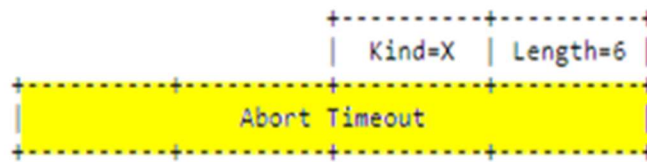
Petition for *Inter Partes* Review

Figure 1: TCP Abort Timeout Option

As discussed in Section VIII.A.1.e.(1), Eggert's SYN+ATO and SYN/ACK+ATO segments are TCP-variant packets, each of which containing an ATO field as idle time period parameter field with metadata (KIND, LENGTH, ATO value) for an idle time period (abort timeout value). For the reasons in Section VIII.A.1, the incorporation of an ATO field in a SYN or SYN-ACK segment results in a TCP-variant packet.

- c) 15(f) sending, from a first node to a second node, the TCP-variant packet in advance of the TCP-variant connection being established to provide the metadata for the idle time period to the second node,

Claim 15, Example 1: The responder (first node) sends a SYN/ACK+ATO segment that provides the ATO to the initiator (second node). EX1004, 3-5. EX1003, ¶¶446-56. For Claim 15, Example 2, the initiator (first node) sends the SYN+ATO that contains the TCP ATO to the responder (second node).

As discussed in element 1(d), the SYN+ATO and SYN/ACK+ATO are both sent in advance of the establishment of the TCP-variant connection. The bits in the ATO field that represent the negotiated timeout value in seconds constitute metadata

Patent No. 9,923,995
Petition for *Inter Partes* Review
for the abort timeout. EX1004, 2.

- d) 15(g) for use by the second node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.

As discussed in Section VIII.A.1.h, the abort timeout used for the connection is a timeout attribute associated with the connection. EX1003, ¶¶457-62.

Claim 15, Example 1: The initiator (second node) uses the value received in the ATO sub-field in the SYN/ACK+ATO segment to modify the abort timeout attribute to be used for the connection by adopting a negotiated timeout that is different than what it proposed initially:

The host that initially proposed the Abort Timeout Option analyzes the next segment it receives from its peer. If the next reply segment does not contain an Abort Timeout Option, the connection MUST use the default abort timeout. If it does, the connection MUST use the abort timeout contained inside the Abort Timeout Option. This can be a different abort timeout than initially proposed, if the peer decided to shorten it.

EX1004, 4.

Claim 15, Example 2: The responder (second node) uses the value received in the ATO sub-field in the SYN+ATO segment to modify the abort timeout attribute to be used for the connection. For example, the responder (second node) may modify, e.g., shorten, the received timeout to a negotiated timeout value that will be used. EX1004, 3.

Patent No. 9,923,995

Petition for *Inter Partes* Review

Using the value received in the data field of the ATO as the abort timeout value is consistent with the '995 Patent: "ITP monitor component 556 may set a timer configured to expire in a time duration identified based on the first idle information received by ITP policy component 550. The identified duration may be longer, shorter, or equal to a duration of the idle time period." EX1001, 17:27-29.

15. Claim 16

Eggert discloses "the apparatus is configured such that modifying the timeout attribute reduces a number of keep-alive signals that are required to be communicated" (claim 16). *See* Section VIII.A.15; EX1003, ¶468-77. In addition, RFC 1122 is incorporated in Eggert as a normative, i.e., indispensable, reference. EX1004, 2, 8; EX1003, ¶476, 151.

According to RFC 1122, a TCP implementation may include keep-alive messages. EX1007, §4.2.3.6, 101-102. Specifically, "[a] 'keep-alive' mechanism periodically probes the other end of a connection when the connection is otherwise idle, even when there is no data to be sent." *Id.* §4.2.3.6, 102. If there is a disconnection, the responder will keep sending keep-alive messages as long as the user timeout has not expired because it is not receiving packets from the initiator. One obvious implementation of Eggert is employing keep-alive packets per RFC 1122. Since the keep-alive packets are sent at a regular interval, shortening the abort timeout will reduce the number of keep-alive signals that are transmitted when there

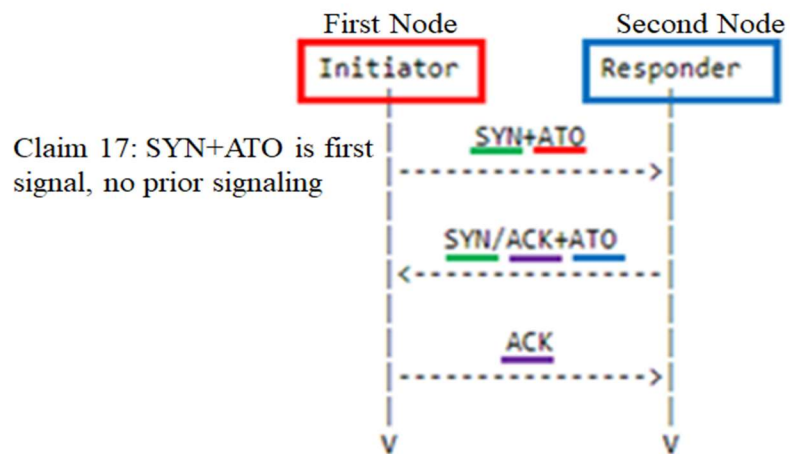
Patent No. 9,923,995

Petition for *Inter Partes* Review

is a disconnection.

16. Claim 17

Eggert discloses “the apparatus is configured such that the TCP-variant packet and the metadata included therewith are sent by the first node to the second node, without any prior signaling received by the first node from the second node” (claim 17). *See* Section VIII.A.14 incorporated by reference; EX1003, ¶¶480-83. In claim 15, second example, the **SYN+ATO** segment is the first message sent to establish the connection.



Claim 15- Second Example
(Ex. 1004, Fig. 2, left side, annotated)

17. Claim 18

a) [18.a]

Eggert discloses “the metadata includes a first value” (claim 18). For example, Eggert discloses that the metadata includes the abort timeout value shown in Fig. 1

Patent No. 9,923,995
 Petition for *Inter Partes* Review
 of Eggert. EX1004, 3; EX1003, ¶¶487-89.

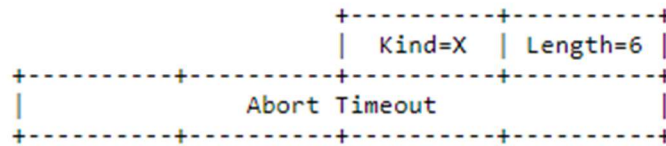


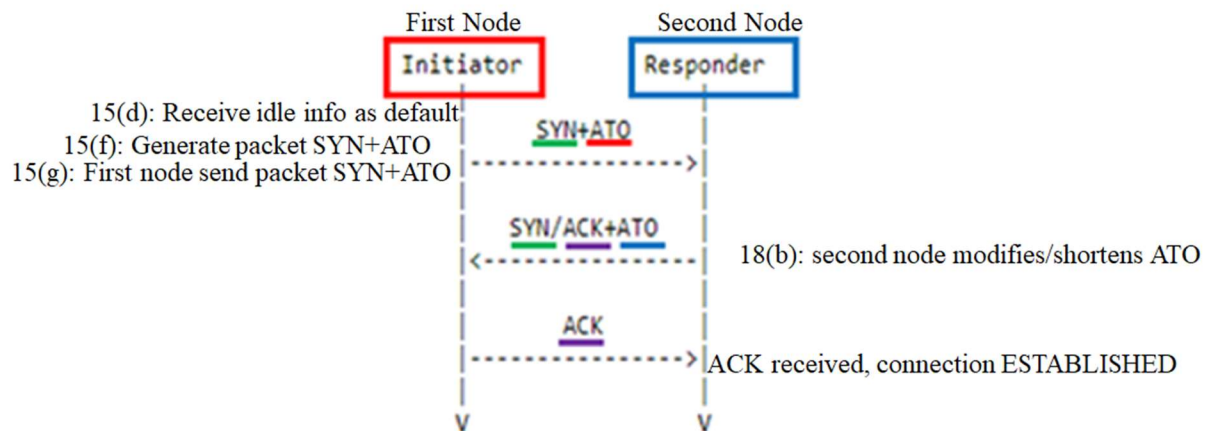
Figure 1: TCP Abort Timeout Option

b) [18.b]

Eggert discloses “the timeout attribute is capable of being modified, by the second node, to include a second value that is different than the first value of the metadata” (claim 18). EX1003, ¶¶490-94.

Claim 15, Example 2: When the responder (second node) receives the ATO, it can shorten the proposed received timeout to a value that is still larger than its default timeout value. EX1004, 3, 6 (“When initiators propose very long timeouts, recipients may reduce the timeout offer to an acceptable length that may still be longer than the default.”).

Patent No. 9,923,995

Petition for *Inter Partes* Review

Claim 15- Second Example
(Ex. 1004, Fig. 2, left side, annotated)

18. Claim 24

Eggert discloses “the apparatus is configured such that the TCP-variant packet and the metadata included therewith are sent by the first node to the second node, in response to receiving, by the first node from the second node, another TCP-variant packet with other metadata” (claim 24). EX1003, ¶541-46.

Claim 15, Example 1: The **SYN/ACK+ATO** segment from the **responder (first node)** is a TCP-variant packet that contains in the ATO sub-field metadata for an abort timeout to be used for the connection and is a response to a first **SYN+ATO** segment received from the **initiator (second node)**.

As discussed in elements 1(f) and 3(a) incorporated by reference, the **SYN+ATO** segment is also a TCP-variant packet that contains as metadata the ATO suggested by the **initiator** for the connection.

Patent No. 9,923,995

Petition for *Inter Partes* Review

B. Ground 2: Claims 1-24 Would Have Been Obvious Over Eggert In View Of Hankinson

To the extent it would not have been obvious based on Eggert alone to implement an apparatus having a non-transitory memory and one or more processors as required by elements 1(a)/15(a) and 1(b)/15(b), these and the other limitations of claims 1-24 would have been obvious to a POSITA based on Eggert in view of Hankinson. EX1003, ¶¶547-579. Specifically, it would have been obvious to install the modified TCP implementation of Eggert in a server computer disclosed in Hankinson that runs Solaris. *Id.* ¶548. Indeed, the motivation comes from Eggert, which discusses Solaris as an example of an operating system that incorporates a TCP implementation that could benefit from the option. *Id.*

A POSITA would have been motivated to store the applications software, TCP protocol software, and/or modified TCP protocol software on a non-transitory memory such as the hard drive of Hankinson so that the software would remain on the server computer even when the server computer is off, and so that the software can readily be used as configured when the server computer is turned on. EX1003, ¶550. Otherwise, the software would have had to be reconfigured every time the server is turned on. *Id.*

Eggert discloses that Solaris allows for connected computers to use different timeouts, even during the ESTABLISHED state of the connection. EX1004, 2. A

Patent No. 9,923,995

Petition for *Inter Partes* Review

POSITA would have been motivated to enable the Solaris computer and other computers to negotiate so that connected nodes use the same timeout values after a connection is established, i.e. in the ESTABLISHED state. EX1004, 3 (“Connections use abort timeouts negotiated with Abort Timeout Options during the ESTABLISHED state only.”); EX1003, ¶551. In addition, adding the TCP ATO to Hankinson would have allowed Hankinson to interoperate with a wider set of computers because, in addition to other computers that supports the TCP ATO, the Solaris server of Eggert in view of Hankinson would also be able to communicate with other computers that only incorporate the standard TCP implementations, without the TCP ATO. EX1004, 4; EX1003, ¶551-53.

A POSITA would have been further motivated to implement Eggert on the computing system of Hankinson to provide key Internet services and protocols described in Hankinson, including web serving capabilities. EX1003, ¶551-52. No other modifications are made to Eggert based on Hankinson. In addition, Hankinson incorporates RFC 793 by reference and explicitly discloses the TCP 3-way handshake for claims 13 and 23. EX1005, [0126] (“TCP is implemented in Thunder OS . . . (RFC) 793, which is incorporated herein by reference”); *id.* [0127], [0133] (“TCP’s ‘three way handshake’.”). All other elements of claims 1-24 are obvious over Eggert as described in Ground 1, which are incorporated by reference here.

1. Claims 1/15

Patent No. 9,923,995

Petition for *Inter Partes* Review

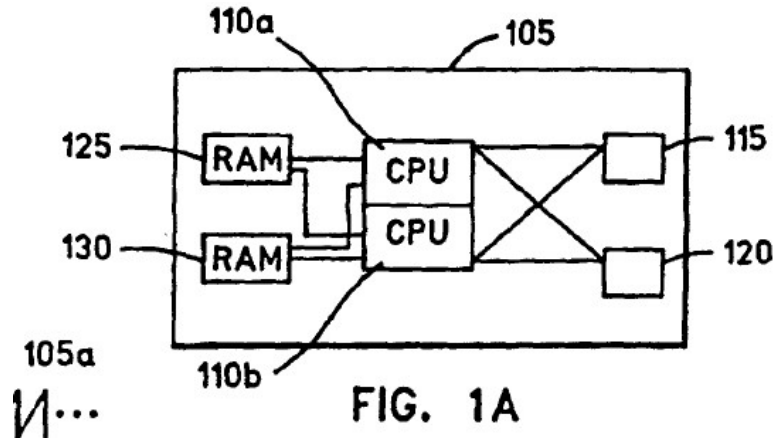
- a) 1(a)/15(a) a non-transitory memory storing instructions;
and

It would have been obvious at the time of the alleged invention to use a server computer that incorporates a non-transitory memory to store the operating system (e.g., Solaris operating system) and TCP implementation (e.g., Solaris's TCP implementation) and the modified TCP implementation (e.g., with TCP ATO) disclosed by Eggert in view of Hankinson. EX1003, ¶558-61. For example, as discussed in Sections VIII.A.1.a-c, Eggert discloses a Solaris computer as an example of a TCP implementation that supports timeouts and would benefit from the use of the TCP ATO. EX1004, 2. Hankinson discloses a computer system that implements an operating system such as Solaris or Linux in firmware and/or software installed on a conventional hard drive, CD-ROM, or just ROM. EX1005, [0033] (“[T]he characteristics of the specific operating system implemented by each member are defined by the specific hardware, firmware (for example ROM), and/or software combination used to implement that particular member.”). Specifically, Hankinson discloses:

For example, the code may be embodied in a signal-bearing medium such as . . . a **CD-ROM**, . . . a **conventional "hard drive"** . . . electronic read-only memory (e.g., **ROM**, EPROM, or EEPROM) . . .

Id. [0157].

Patent No. 9,923,995

Petition for *Inter Partes* Review*Id.* Fig. 1A.

As evidenced by Hankinson, server computers such as Solaris or Linux servers were readily available with a hard disk as a non-transitory memory for storing an operating system such as Solaris, TCP protocol, and applications (e.g., HTTP server). *Id.* [0081] (“Solaris, or any other open source or non open source non real time layer”); *id.* [0123] (“ThunderOS incorporates TCP/IP and Internet server software”); *id.* [0120] (“These key Internet software services and protocols include: HTTP (Hyper Text Transfer Protocol)”).

- b) 1(b)/15(b) one or more processors in communication with the non-transitory memory,

It would have been obvious at the time of the alleged invention to use a server computer that incorporates a processor in communication with the non-transitory memory to load and execute the operating system (e.g., Solaris operating system) and TCP implementation (e.g., Solaris’s TCP implementation) and the modified

Patent No. 9,923,995

Petition for *Inter Partes* Review

TCP implementation (e.g., with TCP ATO) disclosed by Eggert in view of Hankinson. EX1003, ¶563-67.

As evidenced above by Hankinson, server computers such as Solaris or Linux servers were readily available incorporating one or more CPUs in communication with a conventional hard drive, CD-ROM, or just ROM to execute an operating system stored on a hard disk as a non-transitory memory. EX1005, [0033], [0157], [0081]. For example, the CPU executes a program in the ROM 130 to load operating system kernel and other software (e.g., TCP protocol and HTTP application) stored in a hard drive. EX1005, [0156].

2. Claims 11/21

Hankinson does not specify any mechanism to signal the other node that an idle time period has been detected. As discussed in Ground 1 incorporated by reference, Eggert and RFC 793 do not specify any mechanism to signal the other node that an idle time period has been detected. Further, a POSITA implementing the combination of Eggert with Hankinson would not signal the other node. EX1003, ¶568. That is, a POSITA would implement the abort timeout of Eggert “without signaling another one of the first and second nodes that is related to the detection of the idle time period.” *Id.* ¶569.

As discussed in Ground 1 incorporated by reference, the TCP RESET message is not sent when the user timeout expires. EX1006, 77.

Patent No. 9,923,995

Petition for *Inter Partes* Review

3. Claim 12(b)

To the extent it is found that Eggert does not provide an example of an application that operates at the network layer, it would have been obvious for a POSITA to provide an HTTP or web application at the application layer of the host of Eggert in view of Hankinson. EX1003, ¶571-77. Hankinson discloses a ThunderOS that implements HTTP applications (e.g., HTTP session management, Web hosting) on top of TCP. Because Eggert's TCP-variant protocol remains TCP-compliant except for the negotiation that occurs during the handshake, operation of the HTTP application layer will not be impacted by Eggert's protocol. Thus, Hankinson's HTTP applications would operate normally over the Eggert's TCP-variant protocol. As discussed in Ground 1, because the HTTP application resides in the application layer, it is above the transport layer of TCP and the modified TCP of Eggert, both of which are above the internet protocol (IP) layer. For example, Hankinson discloses that the Thunder OS operating system is a network application that implements HTTP protocol to provide key internet services. For instance, Hankinson discloses "Internet applications, which preferably are implemented with a ThunderOS embodiment of the Federated OS." *Id.* [0115].

For increased speed, ThunderOS is implemented such that key Internet services and protocols are implemented directly as part of the distributed operating system. These key **Internet software services** and protocols include:

Patent No. 9,923,995

Petition for *Inter Partes* Review

HTTP (Hyper Text Transfer Protocol)

Id. [0120]; *id.* [0072] (“dispatcher . . . HTTP session management”). Moreover, “[s]ervers implemented with the Federated OS are well suited for many high performance applications, for example, deploying large-scale Internet/intranet applications such as e-commerce, Web hosting.” *Id.* [0032].

4. Claims 13/23

Hankinson explicitly discloses the TCP 3-way handshake required by Claims [13.a]/[23.a]; EX1003, ¶¶578-79. For example, Hankinson discloses:

The creation of a connection (TCP's "three way handshake") has a minimal dependence on the service, and requires only knowledge of which services to associate with given port numbers.

EX1005, [0127]; *id.* [0133]. In addition, the '995 Patent admits that “RFC 793 describes a ‘three-way handshake’ for establishing a TCP connection” and Hankinson incorporates RFC 793 by reference. EX1001, 13:46-48.

C. Ground 3: Claim 16 Would Have Been Obvious Over Eggert In View Of Hankinson and Further In View Of RFC 1122

To the extent the Board does not agree that RFC 1122 is inherently disclosed in Eggert, claim 16 is obvious over Eggert in view of Hankinson further in view of RFC 1122. EX1003, ¶¶580-82. RFC 1122 is incorporated in Eggert as a normative reference. EX1004, 2, 8. Moreover, RFC 1122 updates RFC 793 which is also incorporated in Eggert as a normative reference, and is incorporated by reference in

Patent No. 9,923,995

Petition for *Inter Partes* Review

Hankinson. Thus, a POSITA would have been motivated to combine Eggert and Hankinson and RFC 1122 by incorporating Eggert's ATO into Hankinson, while incorporating into RFC 793 the updates required by RFC 1122. EX1003, ¶582. Indeed, RFC 1122 "is an official specification for the Internet community. It incorporates by reference, amends, corrects, and supplements the primary protocol standards documents relating to hosts." EX1007, 1; Section VIII.A.16 incorporated by reference.

IX. DISCRETIONARY DENIAL UNDER 35 U.S.C. §325(d) WOULD BE INAPPROPRIATE

A. The Prior Art Relied On Herein Is Not The Same Or Substantially Similar To Any Art The Examiner Considered

Eggert is unlike any prior art the Examiner considered and no arguments similar to those contained herein were ever presented to the Office. Patent Owner may assert that RFC 5482, listed on the face of the '995 Patent (but not identified as "cited by examiner"), is similar to Eggert. It is not, and there is no evidence the Examiner even actually considered it.

While RFC 5482 was cited in an IDS filed between a non-final rejection (made purely on double patenting grounds) and notice of allowance, there is no indication that the Examiner considered it. EX1002, 280-284. "[M]ere submission of a reference in an IDS is insufficient for purposes of exercising discretion under § 325(d)." *Samsung Electronics Co., Ltd. v. Seven Networks, Inc.*, IPR2018-01106,

Patent No. 9,923,995

Petition for *Inter Partes* Review

Paper 21, 21 (Nov. 28, 2018); *Fox Factory, Inc. v. SRAM, LLC*, IPR2016-01876, Paper 8, 7-8 (Apr. 3, 2017) (declining to exercise discretion based on §325(d) where the prior art was merely “cited on an IDS [and] there is no evidence that it was considered by the Examiner”); *Alcatel-Lucent USA v. Oyster Optics, LLC*, IPR2018-00070, Paper 14, 15 (May 10, 2018) (same).

Additionally, Eggert’s protocol is fundamentally different from RFC 5482 because, in contrast with RFC 5482, Eggert’s protocol calls for a common negotiated value for the timeout. While Eggert discloses a negotiation protocol, RFC 5482 was purposefully designed such that “an exchange of UTO [user timeout] options between both ends of a connection is not a binding negotiation.” EX1014, 4. Here, claims of the ’995 Patent require that the value of the timeout attribute results from a negotiation or a negotiation protocol.

Eggert also has meaningful disclosures that do not appear in RFC 5482, including section 2.1. Section 2.1 (including Figure 2) of Eggert explicitly requires transmitting the ATO during the SYN and SYN-ACK handshake (the first 2 signals in the three-way handshake), resulting in the exchange of idle timeout information “in advance” of the connection being established as required by all the ’995 Patent challenged claims. EX1004, 5. In contrast, RFC 5482 “does not define a mechanism to negotiate support of the TCP User Timeout Option [UTO] during the three-way handshake.” EX1014, 9. In fact, RFC 5482 allows for exchanging UTO options after

Patent No. 9,923,995

Petition for *Inter Partes* Review

the three-way handshake. EX1014, 10. Therefore, even if the Examiner had considered RFC 5482, these “material differences between the asserted art and the prior art involved during examination” weigh against discretionary denial. *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8, 17-18 (Dec. 15, 2017) (precedential as to §III.C.5, first paragraph) (“Becton Dickinson factors”).

Moreover, no prior art arguments were made at all during prosecution. *See* Section III.B. Instead, the Examiner stated that all claims would be allowed but for a double patenting rejection—later cured by a terminal disclaimer. *Id.* Notably, the IDS citing RFC 5482 was filed after this single rejection, immediately before allowance. RFC 5482 was thus not the basis of any rejections, not discussed in any arguments, and not otherwise relied upon during examination. *See Becton Dickinson* factors.

Thus, under the Board’s two-prong approach, given that no similar art or arguments were considered, the Board need go no further in evaluating the *Becton Dickinson* factors. *Advanced Bionics v. MED-EL*, IPR2019-01469, Paper 6 (Feb. 13, 2020)(precedential). But even if the Board were to find that the Office did consider RFC 5282 (it did not) and even if Eggert were found to be substantially similar to RFC 5282 (it is not), the Office clearly erred in a manner material to the patentability of the challenged claims when it issued the ’995 Patent because the challenged claims are demonstrably unpatentable for the reasons discussed above,

Patent No. 9,923,995

Petition for *Inter Partes* Review

so discretionary denial would be inappropriate for this additional reason. *See Becton Dickinson* factors.

B. The Prior Art Relied On Herein Is Not The Same Or Substantially Similar To Any Art In Unified's IPR Petition

On April 1, Unified Patents filed an IPR against the '995 Patent ("*Unified*") challenging certain claims. IPR2020-00742, 3 (Apr. 1, 2020). Unified certified that no other party exercised control over its petition. *Id.* 1. Thus, Petitioners were unaware of *Unified* and began substantial preparations for filing this petition before *Unified* was filed. Moreover, *Unified* applies different prior art, and did not challenge seven of the claims challenged herein. The *Becton Dickinson* factors weigh against denial.

First, the same or substantially the same art or arguments were not presented in *Unified*. *Unified* presents four grounds based on five references. *Id.* 3-5. There are material differences in the art and grounds presented in both petitions. Notably, the only overlap in prior art with the present petition is a minor discussion of RFC 793. RFC 793 is cited herein only to establish aspects of the TCP protocol unrelated to the invention alleged in the '995 Patent. Further, in *Unified*, RFC 793 is only used as an AAPA secondary reference under Ground 4 challenging claims 25, 26, and 28 not challenged here.

Second, the prior art in this petition is not cumulative of the art in *Unified*. *Unified* relies on U.S. Patent 6,674,713 ("Berg"), whereas this petition relies on

Patent No. 9,923,995

Petition for *Inter Partes* Review

Eggert. Berg discloses a reliable UDP layer that runs on top of the UDP protocol.

Conversely, Eggert discloses a TCP based (not UDP based) protocol.

Third, there are additional facts and evidence presented here that warrant consideration of this art and arguments.

Finally, *Unified* does not challenge claims 8, 11, 13, 21, 23, and 24.

X. DISCRETIONARY DENIAL UNDER 35 U.S.C. §314(a) WOULD BE INAPPROPRIATE

A. *Unified* Provides No Basis For Discretionary Denial

This petition is not a “follow on” petition that the Board should deny under *General Plastics. General Plastic Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19, 16–17 (Sept. 6, 2017)(precedential). None of the non-exclusive factors that the Board considers in exercising discretion on instituting review as to “follow-on” petitions challenging the same patent as challenged previously in an IPR weigh in favor of a denial here. While *Unified* and Petitioners here are different parties, the reasons identified in *Valve* also do not apply. *Valve Corp. v. Elec. Scripting Prods., Inc.*, IPR2019-00062 Paper 11 (Apr. 2, 2019)(precedential). As noted above, *Unified* certified that no other parties had control over its petition. Further, this Petition was filed shortly after *Unified* and before any Preliminary Response. Thus, Petitioners here did not leverage any perceived flaws in *Unified*.

Patent No. 9,923,995

Petition for *Inter Partes* Review

Finally, this Petition is being filed by three consolidated Petitioners demonstrating efficiencies in this review that weigh against denial.

B. The Related Litigation Provides No Basis For Discretionary Denial

There is no trial date set in the related litigation. Jenam filed the related litigations against Samsung and LG on April 3, 2019. EX1008. Jenam asserted two patents, each containing 30 claims, but only identified one claim from each respective patent as infringed in its complaint. Eight months passed after the filing of the complaint before Jenam identified its asserted claims (nearly every claim in each patent), served infringement contentions, and revealed its infringement theory. EX1049. *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11, 11 (Mar. 20, 2020) (noting that “it is often reasonable for a petitioner to wait to file its petition until it learns which claims are being asserted against it in the parallel proceeding”). On February 12, Defendants sought agreement from Jenam to an early claim narrowing. Jenam refused to agree to narrow its asserted claims until July, at the earliest.

In assessing whether to deny institution under §314 based on a parallel litigation, the Board considers the six factors outlined in IPR2020-00019, at 6. *See also NHK Spring v. Intri-Plex Techs.*, IPR2018-00752, Paper 8, (Sept. 12, 2018)(precedential). As a whole, these factors do not favor a discretionary denial.

Specifically, factors two and three strongly weigh against a discretionary

Patent No. 9,923,995

Petition for *Inter Partes* Review

denial. As a result of the delays in the schedule, there has been minimal investment in the case by the district court. There were several continuances of early deadlines due to changes of counsel. EX1048. As a result, discovery has just begun and the parties are just beginning to identify terms for claim construction. EX1048-49. The *Markman* hearing is estimated for some time in October 2020. EX1049. No pretrial deadlines or trial date have been set. EX1049. In *Resideo Techs., Inc. v. Innovation Sciences, LCC*, IPR2019-01306, Paper 19, 11 (Jan. 27, 2020), the PTAB noted the fact that “the district court has not yet set a trial date is a significant factor distinguishing this case from NHK Spring” and noted that “the district court schedule, including the remaining schedule for discovery, does not appear to be firmly set.” *Id.* As a result, the PTAB found that “because there is no trial date set in the parallel litigation, and the schedule continues to change, the schedule of the parallel litigation does not weigh in favor of denying institution under § 314(a).”

Further, with respect to the fourth factor, there is not substantial overlap in the issues before the PTAB and the district court. For example, the district court will consider many more references and combinations not presented herein. In addition, the district court will not consider the combination of Ground 2. *Id.* 12 (“Although Petitioner will challenge claims under § 103 as obvious over Heaton in the parallel litigation, the district court will not consider Heaton’s teachings in view of

Patent No. 9,923,995

Petition for *Inter Partes* Review

Nakamura.”); *id.* 13 (determining “that there is not a substantial overlap in the issues before us with those in the parallel litigation”).

Therefore, the present case is not like *NHK Spring* and efficiency, fairness, and the merits discussed above do not support a discretionary denial.

XI. MANDATORY NOTICES PURSUANT TO 37 C.F.R. §42.8

A. Real Parties-In-Interest under 37 C.F.R. §42.8(b)(1).

Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc., are real parties-in-interest with Petitioners LG Electronics, Inc., LG Electronics U.S.A., Inc. and Google LLC. No other party exercised control or could exercise control over this Petition.

Jenam Tech, LLC purports to be the Patent Owner, but based on publicly available materials, other entities may also be the Patent Owner or real parties-in-interest in the outcome of this petition. Jenam Tech is a Texas LLC, with a registered address of 211 West Tyler Street, Suite C, Longview, Texas. Five other entities also have the same registered address: Oso-IP, Aloft Media, Azure, Stragent, and Majen Tech. Andrew Gordon (also known as George Andrew Gordon) is named a member and registered agent of Jenam Tech, LLC, and holds the same position with respect to four other entities that share the same registered address: Aloft Media, Azure, Stragent, and Majen Tech. With respect to the fifth entity at that address, Oso-IP, Gordon is the registered agent. Kevin Zilka is the sole member of Oso-IP, and is or

Patent No. 9,923,995

Petition for *Inter Partes* Review

has been a member of Aloft Media, Azure, and Stragent. Jenam Tech, LLC acquired the '995 Patent from an entity named Sitting Man, LLC on November 1, 2018.

Petitioners identify these entities and associated individuals in an abundance of caution to assist the Board with respect to identification of any potential conflicts of interest.

B. Related Matters Under 37 C.F.R. §42.8(b)(2).

The '995 Patent is the subject of the following district court litigations: *Jenam Tech, LLC v. LG Electronics, Inc. et al.*, 4:19:cv-00249 (E.D. Tex.) and *Jenam Tech, LLC v. Samsung Electronics Co., Ltd. et al.*, 4:19:cv-00250 (E.D. Tex.). The '995 Patent is the subject of the following IPR: IPR2020-00742.

Patent No. 9,923,995

Petition for *Inter Partes* Review

C. Lead and Back-up Counsel under 37 C.F.R. §42.8(b)(3).

<p>Amanda S. Bonner (Lead Counsel) Registration No. 65,224 Mayer Brown LLP 71 South Wacker Drive Chicago, IL 60606 asbonner@mayerbrown.com</p>	<p>Robert G. Pluta (back-up counsel) Registration No. 50,970 rpluta@mayerbrown.com Luiz Miranda (back-up counsel) Registration No. 74,762 lmiranda@mayerbrown.com MAYER BROWN LLP 71 South Wacker Drive Chicago, IL 60606</p> <p>Baldine Paul (back-up counsel) Registration No. 54,369 bpaul@mayerbrown.com Mayer Brown LLP 1999 K Street, NW Washington, D.C. 20006</p>
---	--

D. Service Information under 37 C.F.R. §42.8(b)(4).

Please direct all correspondence to lead counsel at the address identified above. Petitioners consent to electronic service on counsel of record.

XII. GROUNDS FOR STANDING

Petitioner certifies that the '995 Patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting a *inter partes* review of the '995 Patent. 37 C.F.R. §42.104(a).

XIII. PAYMENT OF FEES

A fee of \$37,100 charged to Deposit Account 130019 was paid with the filing this petition. 37 C.F.R. §§42.15(a), 42.103. Should any further fees be required by

Patent No. 9,923,995

Petition for *Inter Partes* Review

the present Petition, the Board is hereby authorized to charge the above-referenced Deposit Account.

XIV. STATEMENT OF PRECISE RELIEF REQUESTED

Petitioner requests IPR and cancellation of claims 1-24 because they are unpatentable under 35 U.S.C. §103 based on the foregoing grounds.

Date: April 17, 2020

Respectfully submitted,

By:/Amanda Streff Bonner/
Amanda Streff Bonner
Registration No. 65,224
Mayer Brown LLP
71 S. Wacker Drive
Chicago, IL 60606
Telephone: 312-701-8645
Facsimile: 312-701-7711
asbonner@mayerbrown.com

Counsel for Petitioners

Patent No. 9,923,995

Petition for *Inter Partes* Review

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§42.6 and 42.105, I hereby certify that on April 17, 2020, a copy of the attached PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 9,923,995, together with all exhibits, and all other papers filed therewith, including a Power of Attorney on behalf of Petitioners, was served via Priority Mail Express on the attorneys of record for the patent owner at the following address:

Attorney of Record for Patent Owner

Patrick Caldwell (Reg. No. 44580)

G. Gordon (Reg. No. 64517)

P.O. Box 59655

Dept. SVIPGP

Dallas, TX 75229

Date: April 17, 2020

Respectfully submitted,

By:/Amanda Streff Bonner/

Amanda Streff Bonner

Registration No. 65,224

Mayer Brown LLP

71 S. Wacker Drive

Chicago, IL 60606

Telephone: 312-701-8645

Facsimile: 312-701-7711

asbonner@mayerbrown.com

Counsel for Petitioners

CERTIFICATION PURSUANT TO 37 C.F.R. § 42.11

Pursuant 37 C.F.R. § 42.11, the undersigned certifies that this PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 9,923,995, and accompanying evidence are not being presented for an improper purpose and that all legal contentions, allegations, and denials are warranted and have evidentiary support.

Date: April 17, 2020

Respectfully submitted,

By:/Amanda Streff Bonner/
Amanda Streff Bonner
Registration No. 65,224
Mayer Brown LLP
71 S. Wacker Drive
Chicago, IL 60606
Telephone: 312-701-8645
Facsimile: 312-701-7711
asbonner@mayerbrown.com

Counsel for Petitioners

CERTIFICATION PURSUANT TO 37 C.F.R. § 42.24(d)

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that this PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 9,923,995, complies with the type-volume limitation of 37 C.F.R. § 42.24(a). The word count application of the word processing program used to prepare this paper states that the paper contains 13,983/14,000 words, excluding the parts of the paper exempted by 37 C.F.R. §§ 42.24(a)(1).

Date: April 17, 2020

Respectfully submitted,

By:/Amanda Streff Bonner/
Amanda Streff Bonner
Registration No. 65,224
Mayer Brown LLP
71 S. Wacker Drive
Chicago, IL 60606
Telephone: 312-701-8645
Facsimile: 312-701-7711
asbonner@mayerbrown.com

Counsel for Petitioners

EXHIBIT 3

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

JENAM TECH, LLC,
Patent Owner

Patent No. 10,075,564

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,075,564**

Petition for *Inter Partes* Review
Patent No. 10,075,564

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	1
III.	PAYMENT OF FEES	2
IV.	GROUND FOR STANDING.....	3
V.	PRECISE RELIEF REQUESTED	3
A.	Claims for Which Review Is Requested	3
B.	Statutory Grounds of Challenge.....	3
VI.	LEVEL OF ORDINARY SKILL	5
VII.	OVERVIEW OF THE '564 PATENT	5
VIII.	CLAIM CONSTRUCTION	9
IX.	DETAILED EXPLANATION OF GROUNDS.....	12
A.	Ground 1: The Combination of <i>Wookey</i> and <i>Eggert</i> Renders Obvious Claims 16, 22, and 23	12
1.	Claim 16.....	12
2.	Claim 22.....	61
3.	Claim 23.....	71
X.	EGGERT IS A PRINTED PUBLICATION	75
XI.	DISCRETIONARY DENIAL IS NOT APPROPRIATE	76
A.	<i>Eggert</i> is Not Cumulative.....	76
B.	The Related Litigation Provides No Basis For Discretionary Denial	78
XII.	CONCLUSION.....	87

Petition for *Inter Partes* Review
Patent No. 10,075,564

LIST OF EXHIBITS

EX1001	U.S. Patent No. 10,075,564
EX1002	Declaration of Bill Lin, Ph.D.
EX1003	Curriculum Vitae of Bill Lin, Ph.D.
EX1004	Prosecution History of U.S. Patent No. 10,075,564
EX1005	U.S. Pre-Grant Publication No. 2007/0171921 to Wookey <i>et al.</i>
EX1006	L. Eggert, TCP Abort Timeout Option, draft-eggert-tcm-tcp-abort-timeout-option-00, Network Working Group, Internet-Draft (April 14, 2004)
EX1007	U.S. Patent No. 6,981,048 to Abdolbaghian
EX1008	DARPA RFC 793 TRANSMISSION CONTROL PROTOCOL
EX1009	RFC 1122 “Requirements for Internet Hosts -- Communication Layers”
EX1010	U.S. Patent No. 9,923,995
EX1011	RESERVED
EX1012	U.S. Patent No. 6,212,175 to Harsch
EX1013	U.S. Patent No. 6,665,727 to Hayden
EX1014	U.S. Patent No. 7,636,805 to Rosenberg
EX1015	U.S. Patent No. 6,584,546 to Kavipurapu
EX1016	IETF RFC 5482
EX1017	<i>Jenam Tech, LLC’s</i> First set of Infringement Contentions regarding U.S. Patent No. 10,075,564 (August 21, 2020)

Petition for *Inter Partes* Review
Patent No. 10,075,564

EX10181	<i>Jenam Tech, LLC v. LG Electronics, Inc. et al</i> , Case No. 4:19-cv-00249, ECF No. 1 (E.D. Tex. Apr. 3, 2019)
EX1019	Declaration of Alexa Morris for Eggert
EX1020	U.S. Pre-Grant Publication No. 2004/0093376 to De Boor <i>et al.</i>
EX1021	U.S. Patent No. 7,535,913 to Minami <i>et al.</i>
EX1022	U.S. Pre-Grant Publication No. 2005/0204013 to Raghunath <i>et al.</i>
EX1023	U.S. Pre-Grant Publication No. 2007/0005804 to Rideout
EX1024	U.S. Pre-Grant Publication No. 2004/0098748 to Bo <i>et al.</i>
EX1025	IETF RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1
EX1026	U.S. Patent No. 8,259,716 to Diab (“ <i>Diab</i> ”)
EX1027	Bova et al., RELIABLE UDP PROTOCOL <draft-ietf-sigtran-reliable-udp-00.txt> 25 February 1999
EX1028	U.S. Patent No. 6,674,713 to Berg et al. (“ <i>Berg</i> ”)
EX1029	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Joint Submission of Proposed Amended Scheduling Order, ECF No. 57 (W.D. Tex. Mar. 4, 2021)
EX1030	Thirteenth Supplemental Order Regarding Court Operations Under the Exigent Circumstances Created By the COVID-19 Pandemic, The United States District Court for the Western District of Texas (February 2, 2021)
EX1031	<i>Digital Retail Apps, Inc. v. H-E-B, LP</i> , Case No. 6:19-cv-00167, Joint Stipulation and Order Postponing Trial, ECF No. 182 (W.D. Tex. Jan. 13, 2021)
EX1032	<i>Jenam Tech, LLC’s</i> Preliminary Narrowing of Claims (October 20, 2020)
EX1033	<i>Jenam Tech, LLC’s</i> Correspondence Regarding Narrowing of Claims (December 4, 2020)

Petition for *Inter Partes* Review
Patent No. 10,075,564

EX1034	Judge Alan D. Albright's Case Statistics By Year (Retrieved from DocketNavigator on March 9, 2021)
EX1035	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Order Granting Motion to Stay Case, ECF No. 58 (W.D. Tex. Mar. 10, 2021)

Petition for *Inter Partes* Review
Patent No. 10,075,564

I. INTRODUCTION

Google LLC (“Petitioner”) requests *inter partes* review (“IPR”) of claims 16, 22, and 23 (“the challenged claims”) of U.S. Patent No. 10,075,564 (“the ’564 patent”) (EX1001) assigned to Jenam Tech, LLC (“Patent Owner” or “PO”). For the reasons below, the challenged claims should be found unpatentable and canceled.

II. MANDATORY NOTICES

Real Parties-in-Interest: Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies the following as the real parties-in-interest: Google LLC.¹

Related Matters: The ’564 patent is asserted in the following civil action: *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.). The ’564 patent was previously asserted in the following civil action: *Jenam Tech, LLC v. Samsung Electronics Co., Ltd.*, Case No. 4:20-cv-00279 (E.D. Tex.) (voluntarily dismissed).

The ’564 patent claims priority to U.S. Patent Application No. 15/694,802, filed on September 3, 2017, and issued as U.S. Patent No. 9,923,995 (“the ’995

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

Petition for *Inter Partes* Review
Patent No. 10,075,564

patent”). (EX1001, Cover.) The ’995 patent is subject to the following instituted IPRs: *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845 (“Google -845 IPR”); and *Unified Patents, LLC v. Jenam Tech, LLC*, IPR2020-00742 (“Unified -742 IPR”).

Petitioner is concurrently filing another IPR petition challenging the ’564 patent.² Petitioner is also concurrently filing IPR petitions challenging U.S. Patent Nos. 10,075,565, and 10,375,215 which are also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.).

Counsel and Service Information: Lead counsel: Naveen Modi (Reg. No. 46,224). Backup counsel: (1) Joseph E. Palys (Reg. No. 46,508), (2) Quadeer A. Ahmed (Reg. No. 60,835), (3) Jason Heidemann (Reg. No. 77,880).

Service information is Paul Hastings LLP, 2050 M Street NW, Washington, DC 20036, Tel.: 202.551.1700, Fax: 202.551.1705, email: PH-Google-Jenam-IPR@paulhastings.com. Petitioner consents to electronic service.

III. PAYMENT OF FEES

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-2613.

² Petitioner concurrently submits herewith its Notice Regarding Multiple Petitions.

Petition for *Inter Partes* Review
Patent No. 10,075,564

IV. GROUNDS FOR STANDING

Petitioner certifies that the '564 patent is available for review, and Petitioner is not barred or estopped from requesting review on the grounds identified herein.

V. PRECISE RELIEF REQUESTED

A. Claims for Which Review Is Requested

Petitioner respectfully requests review of claims 16, 22, and 23 (“the challenged claims”) and cancellation of those claims as unpatentable given the following ground.

B. Statutory Grounds of Challenge

Ground 1: Claims 16, 22, and 23 are rendered obvious under 35 U.S.C. § 103(a) over U.S. Publication No. 2007/0171921 to Wookey *et al.* (“*Wookey*”) (EX1005) in view of Eggert, TCP Abort Timeout Option (“*Eggert*”) (EX1006).³

The '564 patent issued from Application No. 15/915,047, filed on March 7, 2018, which is a continuation of Application No. 15/694,802, filed on September 3, 2017, which is a continuation-in-part of Application No. 14/667,642, filed on March 24, 2015, which is a continuation-in-part of Application No.

³ Other references discussed herein are provided to show the state of the art at the time of the alleged invention of the '564 patent.

Petition for *Inter Partes* Review
Patent No. 10,075,564

13/477,402, filed on May 22, 2012, which is a continuation of Application No. 12/714,454 (“the ’454 application”), filed on February 27, 2010. (EX1001, 1:8–28.)

For purposes of this proceeding only, Petitioner assumes, without conceding, the earliest effective filing date of the ’564 patent is February 27, 2010, (filing date of the ’454 application).

Wookey published on July 26, 2007, from an application filed on November 14, 2006 (EX1005, Cover) and thus qualifies as prior art at least under 35 U.S.C. § 102(b) (pre-AIA).

Eggert, which is an Internet Engineering Task Force (IETF) Internet-Draft (or “ID”) working document, was published on April 14, 2004. (EX1006, 1; *see also infra* Section X.) As confirmed by the declaration of Alexa Morris, Managing Director of IETF, *Eggert* was published, disseminated, and reasonably available to the public by April 15, 2004. (EX1019 ¶¶9–10; *see also infra* Section X.) Thus, *Eggert* qualifies as prior art at least under 35 U.S.C. § 102(b) (pre-AIA).

The above references were not considered during prosecution of the ’564 patent (EX1001, Cover (“References Cited”); *see also generally* EX1004) and are not the same or substantially similar to any art previously presented to the Office. (*See infra* Section XI.A.)

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art at the time of the alleged invention of the '564 patent ("POSITA") would have had an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking. (EX1002 ¶¶16–18.)⁴ More education can supplement practical experience and vice versa. (*Id.*)

VII. OVERVIEW OF THE '564 PATENT

The '564 patent is directed to networking, and in particular, to the sharing of information for detecting an idle TCP connection. (EX1001, 2:26–28, 7:29–50.) Figure 7 illustrates such a process.

⁴ Petitioner submits the declaration of Dr. Bill Lin (EX1002), an expert in the field of the '564 patent. (EX1002 ¶¶3–15; EX1003.)

Petition for *Inter Partes* Review
 Patent No. 10,075,564

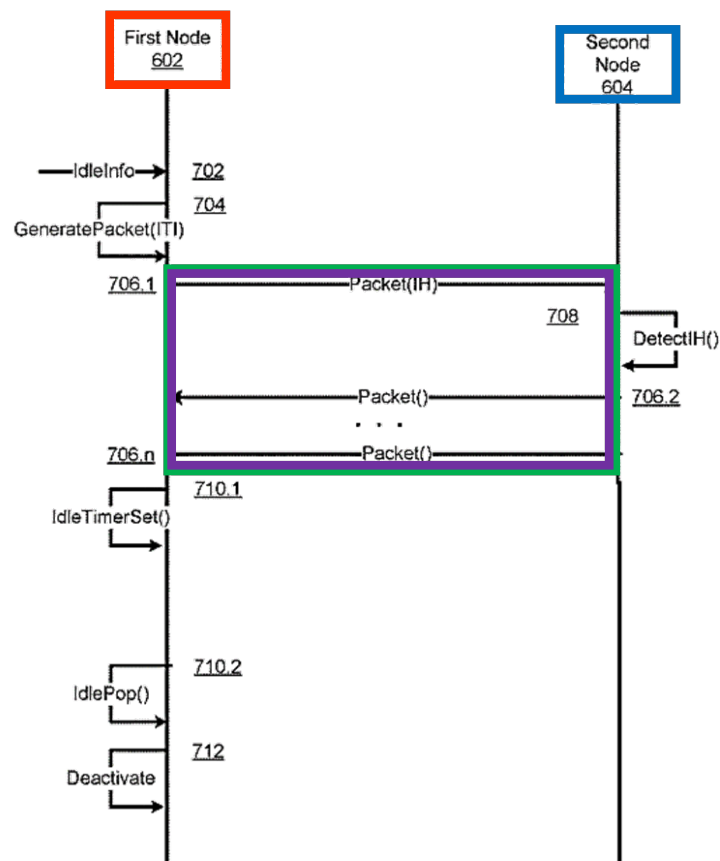


Fig. 7

(*Id.*, FIG. 7 (annotated); EX1002 ¶64.)

First, the **first node 602** receives a message 702 that identifies idle information representing a duration for an idle time period (“ITP”). (EX1001, 10:55–57.) Message 702 may take various forms, such as “a message received via a network.” (*Id.*, 10:57–63.) The idle information “may include and/or identify a duration of time for detecting an idle time period.” (*Id.*, 11:37–41.) The duration

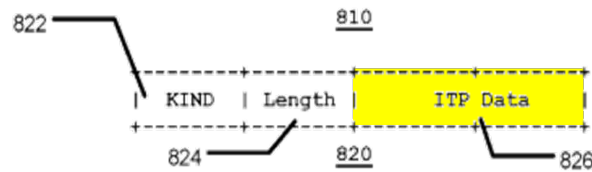
Next, the **first node** “generat[es] a TCP packet including an ITP header based on received idle information.” (EX1001, 15:4–7.) The **TCP options field** of a TCP packet may store this ITP header. (*Id.*, 14:24–29.) Figure 8 below, “adapted from RFC 793,” illustrates this configuration. (*Id.*, FIG. 8, 5:28–30, 14:46–49.)



7

Petition for *Inter Partes* Review
 Patent No. 10,075,564

header can also be “in structures and locations [in a TCP packet] other than those specified for TCP options in RFC 793” (EX1001, 14:55–58).



Figures are adapted from RFC 793

Fig. 8

(*Id.*, FIG. 8 (cropped and annotated); EX1002 ¶¶66–67.)

The ITP header is exchanged during the three-way handshake. (EX1001, 13:49–52.) For example, as shown in Figure 7, the **first node** transmits a message 706.1, i.e., a TCP packet including an ITP header (IH) that contains ITP information, to the **second node 604**. (*Id.*, 15:29–31.) Message 708 exemplifies the **second node**’s detection of the IH in the received TCP packet. (*Id.*, 20:50–53; EX1002 ¶67; *see also id.* ¶¶68–71.)

The challenged claims recite limitations relating to features discussed above. However, all the limitations in the challenged claims were known in the prior art and are obvious. (See Section IX; EX1002 ¶72; *see also id.* ¶¶19–63 (discussing technology background), citing, e.g., EXS. 1007–1009, 1012–1015, 1021–1028.)

VIII. CLAIM CONSTRUCTION

Under the applicable standard in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), claim terms are typically given their ordinary and customary meanings as understood by a POSITA at the time of the invention based on the claim language, specification, and the prosecution history of record. *Phillips*, 415 F.3d at 1313; *see also id.* at 1312–16. The Board, however, only construes the claims when necessary to resolve the controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper No. 11 at 16 (Aug. 14, 2015) (citing *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)). Petitioner believes that no express constructions of any claim terms are necessary to assess whether the prior art reads on the challenged claims.⁵ (EX1002 ¶73.)

⁵ Petitioner reserves all rights to raise claim construction and other arguments, including challenges under 35 U.S.C. §§ 101 or 112, in district court as relevant to those proceedings. *See, e.g., Target Corp. v. Proxicom Wireless, LLC*, IPR2020-00904, Paper 11 at 11–13 (November 10, 2020). A comparison of the claims to any accused products in litigation may raise controversies that are not presented here given the similarities between the references and the patent.

Petition for *Inter Partes* Review
Patent No. 10,075,564

The challenged claims recite the terms “TCP-variant packet” (claim 16, EX1001, 24:40-47) and “TCP-variant connection” (claims 16, 22, 23 (*e.g.*, *id.*, 24:38–39)), which are also claimed in the related ’995 patent at-issue in the Unified -742 IPR and Google -845 IPR (*see supra* Section II). The ’564 patent and the ’995 patent share a common specification. (*Compare* EX1001 *with* EX1010.)

The ’564 patent discloses that the “TCP-variant” protocol varies slightly from TCP by adding an idle timeout option using the format for new TCP options in a TCP header. (EX1001, 14:24–54, 15:4–7, FIG. 8; *supra* Section VII.)

Like the Board noted for the ’995 patent, the ’564 patent’s specification “provides little guidance as to [the] meaning [of ‘TCP-variant connection’].” Google -845 IPR, Paper 16 (Institution Decision) at 17 (October 8, 2020); Unified -742 IPR, Paper 11 (Institution Decision) at 5 (October 8, 2020). (*See generally* EX1001.)⁶

Eggert’s timeout option is a similar variation on TCP as disclosed in the ’564 patent and thus discloses the term “TCP-variant.” (*See, e.g., infra* Sections IX.A.1.c–d.)

⁶ While the ’564 patent generically refers to “variant of the current TCP” (EX1001, 14:60–61), the term “TCP-variant” is only found in the Summary and claims.

Petition for *Inter Partes* Review
 Patent No. 10,075,564

As shown below, *Eggert*'s timeout option (right) uses the same TCP structure as the ITP field illustrated in Figure 8 of the '564 patent (left), including a data portion (YELLOW) containing the timeout value.

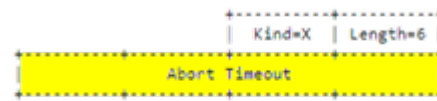
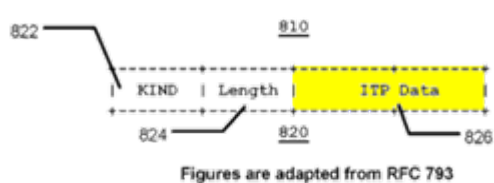


Figure 1: TCP Abort Timeout Option

EX1001, FIG. 8 (annotated)

EX1006, FIG. 1 (annotated)

(EX1002 ¶¶93, 140, 185–186; *infra* Section IX.A.1.d.) Moreover, the Board preliminarily found that *Eggert* discloses these TCP-variant limitations in the Google -845 IPR. *See, e.g.*, Google -845 IPR, Paper 16 at 20–23.

Thus, given that the '564 patent offers no descriptions that would require a special meaning of these terms and that the prior art discloses these features under any reasonable interpretation, the Board need not construe the TCP-variant terms. Instead, as Petitioner does, the Board should apply the plain meaning of the terms “TCP-variant packet” and “TCP-variant connection” like the Board has applied in the Unified -742 IPR and Google -845 IPR. (*See infra* Sections IX.A.1.c–d).

IX. DETAILED EXPLANATION OF GROUNDS**A. Ground 1: The Combination of *Wookey* and *Eggert* Renders Obvious Claims 16, 22, and 23****1. Claim 16****a) A non-transitory computer readable medium, comprising:**

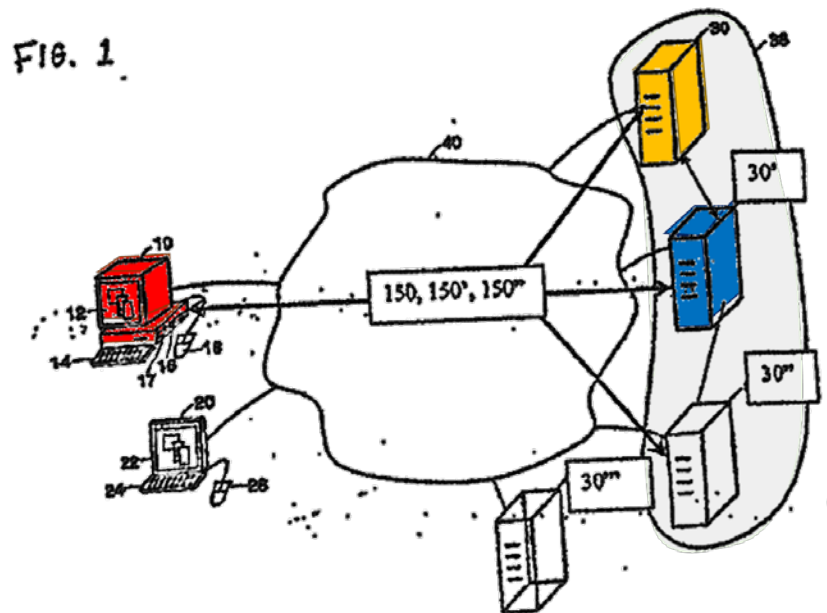
To the extent limiting, *Wookey* discloses the limitations of this preamble. (EX1002 ¶¶94–106.) For example, *Wookey* discloses a **remote machine 30** that includes a main memory unit 104 and cache memory 140 (individually or collectively, the claimed “non-transitory computer readable medium”). (*Id.* ¶95; *see also* EX1005 ¶[0002], Abstract.)

Figures 1, 2C, and 3D below show an exemplary environment in which a **client machine 10**, **remote machine 30**, and **remote machine 30'**, amongst other components, are employed to permit a **client machine 10** access to a computer resource.⁷ (*E.g.*, EX1005 ¶¶[0016], [0020], [0196]; EX1002 ¶¶74–83 (*Wookey*

⁷ *Wookey*'s disclosures (including its figures) include certain typographical errors when referencing its client machines (referring to them interchangeably as 10, 10' or 20). (EX1002 ¶100; EX1005 ¶¶[0135]–[0136], [0152]–[0154], [0192]–[0196], [0207]–[0216], FIGS. 1, 2C, 3D.) For example, the bottom figure in Fig. 2C

Petition for *Inter Partes* Review
 Patent No. 10,075,564

overview).). *Wookey* discloses that the client and remote machines are “typical computers” (EX1005 ¶[0021], FIGS. 1A–1B) including personal computers and wireless devices (*id.* ¶[0171]), such as mobile telephone or other portable telecommunication devices (*id.* ¶¶[0172]–[0173]).



erroneously shows client machine “20,” but the corresponding disclosure for both figures in Fig. 2C describes client machine “10.” (EX1005 ¶¶[0192]–[0196].) Moreover, in context of *Wookey*’s disclosures, a POSITA would have understood that any teachings regarding one client machine 10/10’/20 in *Wookey* apply equally to the other client machines. (EX1002 ¶100.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

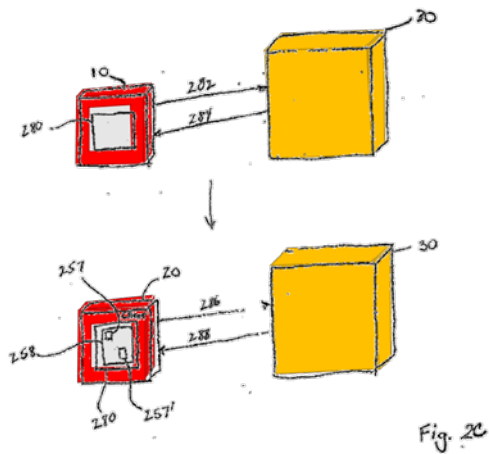
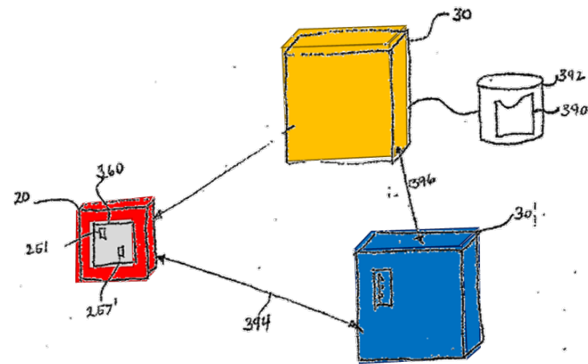


Fig. 3D



(EX1005, FIGS. 1 (annotated), 2C (annotated), 3D (annotated) EX1002 ¶¶96–99.)

Figures 1A and 1B below depict typical computer architectures for both client and remote machines. (EX1005 ¶[0021].)

FIG. 1A

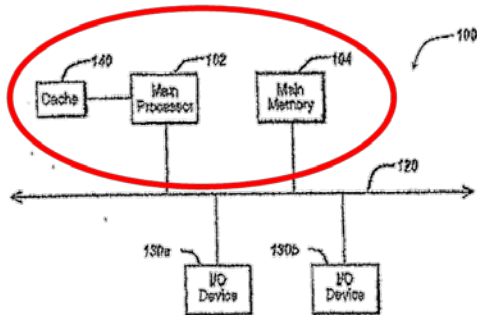
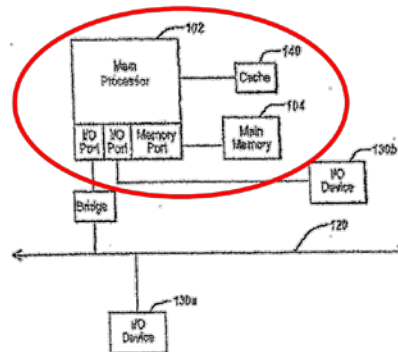


FIG. 1B



(*Id.*, FIGS. 1A and 1B (annotated); EX1002 ¶101.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

Each computer 100, as shown above, includes a central processing unit (CPU) 102, main memory unit 104, and cache memory 140. (EX1005 ¶¶[0161], [0163].)⁸ The CPU 102 “responds to and processes *instructions fetched from* the main memory unit 104.”⁹ (*Id.* ¶[0162]; *see also id.* ¶¶[0554], [0556], [0802], [1121], FIG. 33.) The main memory unit 104 can include one or more memory chips (e.g., SRAM) that store data accessed by CPU 102. (*Id.* ¶[0163].) The cache memory 140 is a fast memory type that acts as a buffer between RAM and the CPU. (*Id.* ¶[0165].) As was known in the art at the time, cache memory stores both data and instructions accessible to the CPU, and thus *Wookey*’s cache 140 would necessarily store instructions accessible to the CPU. (EX1002 ¶¶102–103; EX1015, 1:15–39.) Accordingly, cache memory 140 and main memory unit 104 of the **remote machine 30** are each a non-transitory memory device and thus, individually or collectively, disclose the claimed non-transitory computer medium. (EX1002 ¶¶104–106; *infra* Sections IX.A.1.b–f.)

⁸ *Wookey* uses the terms “Main Processor 102,” “microprocessor 102,” “processor 102,” and “central processing unit 102” interchangeably. (EX1005 ¶¶[0161], [0163]–[0164], FIGS. 1A–B; EX1002 ¶102 n.4.)

⁹ Emphasis added unless otherwise noted.

b) **Claim 16.b**

Wookey discloses this limitation, which is addressed below in two parts.
(EX1002 ¶107.)

- (1) code for being communicated to a remote client node, [] where the code, when used by the client node, results in the client node operating to:¹⁰**

Wookey discloses limitation 16.b(1). (EX1002 ¶¶108–116.) For instance, *Wookey* discloses that encoded Uniform Resource Locators (URLs) (“code”) associated with icons on an HTML page 288 (where the HTML page 288 is also “code”)¹¹, which are stored in **remote machine 30**’s memory, are communicated to

¹⁰ The ’564 patent specification does not provide any disclosure describing the claimed “code” features in the claim. (EX1002, ¶108 n.7.) Therefore, while Petitioner addresses such features with respect to the asserted prior art, Petitioner does not concede that the ’564 patent provides adequate disclosures of such claimed features.

¹¹ Petitioner’s analysis regarding the “code” limitation equally applies whether the “code” is the encoded URLs in the HTML page or the HTML page itself (which includes the encoded URLs). (EX1002 ¶109 n.8.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

client machine 10 (“remote client node”/“client node”). (EX1005 ¶¶[0192]–[0193].)¹² When used by **client machine 10**, this code causes **client machine 10** to initiate a process for establishing a second connection with another **remote machine 30**’ as discussed below for limitations 16.c–e. (*Id.* ¶¶[0196], [0207], [0213]; EX1002 ¶109; *infra* Sections IX.A.1.c–e.)

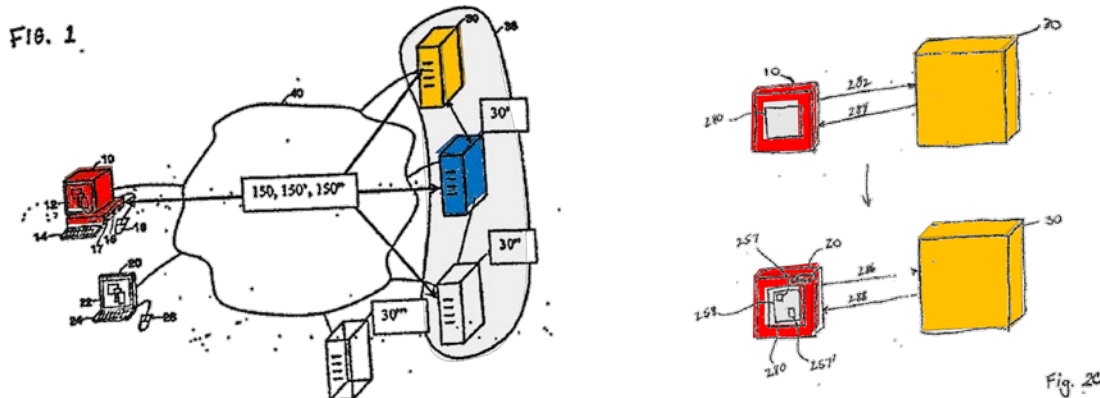
Client machine 10 is “a remote client node,” as claimed. (EX1002 ¶¶110–112.) For example, **client machine 10** may access a computer resource provided by **remote machine 30**. (*E.g.*, EX1005 ¶¶[0135]–[0136], [0154], [0171].) Figure 1 exemplifies “an environment in which a client machine 10, 10’ accesses a computing resource provided by a remote machine” (*id.* ¶[0135]) “over [a] communications link 150” (*id.* ¶[0154]). (*See also id.* ¶[0245].) Further, *Wookey* explains that **client machine 10** executes a web browser application 280 (*id.* ¶[0192]) that “*transmits* a request 282 to access a Uniform Resource Locator (URL) address corresponding to an HTML page residing on” **remote machine 30** (*id.* ¶[0193]).¹³ (*Id.* ¶¶[0021],

¹² *Wookey* uses the term “machine” to describe what the ’564 patent refers to as a “node.” (EX1005 ¶¶[0153], [0301]); EX1002 ¶109.)

¹³ *Wookey*’s reference to “remote machine 10” in the first sentence of paragraph [0193] is a typographical error. In the same paragraph, *Wookey* identifies the same

Petition for *Inter Partes* Review
 Patent No. 10,075,564

[0024], [0213], FIG. 2C.) Accordingly, **client machine 10** is *remote* to **remote machine 30** because it may access a resource over some geographical distance through a network connection. (EX1002 ¶¶111–112.)



(EX1005, FIGS. 1, 2C (each annotated); EX1002 ¶¶110–112.)

Wookey discloses “code for being communicated to a remote client node...” like that recited in limitation 16.b.1. (EX1002 ¶113.) For example, **remote machine 30** “prepares and transmits to the client machine 10 an HTML page 288 that includes a Resource Neighborhood window 258 in which appears graphical icons 257, 257’ representing resources to which the client machine 10 has access.” (EX1005 ¶[0196]; *see also id.* ¶¶[0215]–[0216].) The HTML page, which is communicated

remote machine as “the remote machine 30,” consistent with Figure 2C. (EX1002 ¶112 n.9; EX1005 ¶¶[0193], [0195]–[0196], FIGS. 2C, 3D.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

to **client machine 10**, is stored in **remote machine 30**'s memory. (*Id.* ¶[0207].) Thus, a POSITA would have recognized that the HTML page 288 (“code”), which includes encoded Uniform Resource Locators (URLs) (also “code”) associated with icons on the HTML page as explained below, is communicated by **remote machine 30** to **client machine 10**. (EX1002 ¶113.)

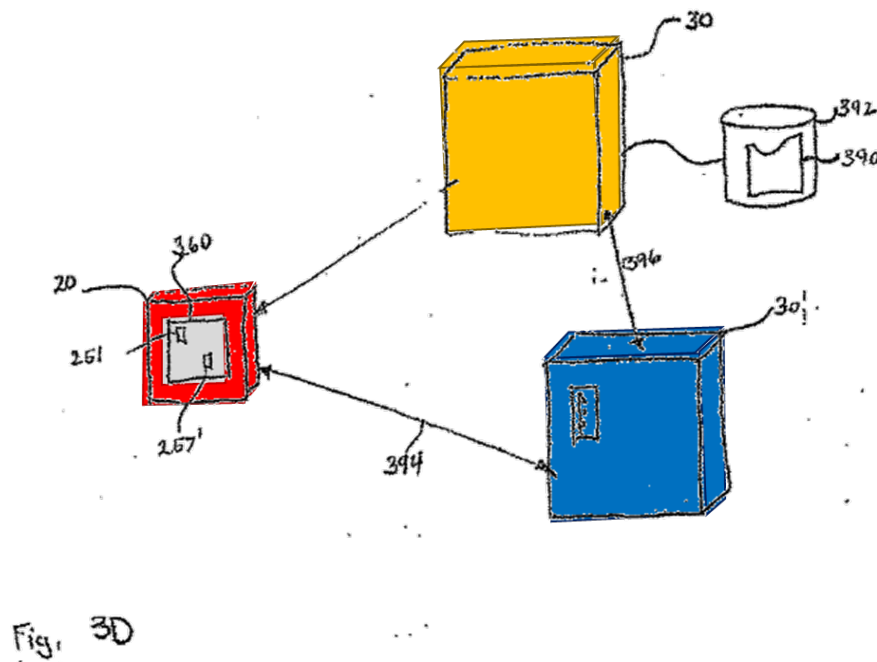
The HTML page 288 includes encoded URLs associated with icons (e.g., 257, 257'). These encoded URLs are the claimed “code, when used by the client node, results in the client node operating to” initiate a process for establishing a second connection with a second **remote machine 30'** and perform the steps outlined in limitations 16.c–e. (*Id.* ¶114; *infra* Sections IX.A.1.c–e.)

In particular, “[e]ach icon 257, 257' *is associated with an encoded URL that specifies: the location of the resource . . . ; a launch command associated with the resource; and a template identifying how the results of accessing the resource should be displayed.*” (EX1005 ¶[0216].) *Wookey* also discloses that each encoded URL “*contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (*Id.*) Thus, when a user *clicks* an icon 257, 257' corresponding to a resource from the displayed HTML page 288, **client machine 10** uses the encoded URL associated with the clicked icon to initiate the

Petition for *Inter Partes* Review
 Patent No. 10,075,564

process for establishing a second connection (e.g., connection 394 in Figure 3D below) with a second **remote machine 30'**. (*Id.*; see also *id.*, [0026]; EX1002 ¶114.)

Accordingly, the encoded URLs are code that, when used by **client machine 10**, results in **client machine 10** operating to perform specific processes because such URLs are instructions that, when invoked, command **client machine 10** to access resources from one or more locations (e.g., one or more remote machines). (EX1002 ¶115.)



(EX1005, FIG. 3D (annotated); EX1002 ¶114.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

The above understanding is consistent with how the '564 patent describes its “code” as “includ[ing] one or more links.” (EX1001, cl. 30 (27:6).) Indeed, *Wookey*’s encoded URLs similarly include a link (e.g., specifying the remote resource location and launch command to connect to the resource). (EX1002 ¶115.)

Moreover, the HTML page 288 (including the encoded URLs) also discloses the claimed “code” because an HTML page was known to comprise HTML code used to define, e.g., what is displayed on the screen of the client node. (*Id.* ¶116; EX1020, ¶¶[0022], [0024], [0026], [0079].) The HTML code used to display information relating to HTML page 288 would include the icons associated with the URLs that the client node uses to access the desired resource, as further discussed below for limitation 16.c. Accordingly, HTML page 288 includes code. (*Id.* ¶116.) This understanding (e.g., HTML page includes code) is consistent with PO’s interpretation of this limitation in its infringement positions for the '564 patent. (EX1017, 45 (“code (e.g., HTML pages, etc.)”).)¹⁴

¹⁴ Petitioner’s references herein to PO’s infringement allegations are not indicative of any concession to PO’s allegations that any accused instrumentality infringes any claim limitation.

Petition for *Inter Partes* Review
 Patent No. 10,075,564

- (2) [] the client node including one or more processors in communication with a non-transitory memory;

Wookey discloses limitation 16.b(2). (EX1002 ¶¶117–119.) For example, *Wookey* explains that **client machine 10** (“client node”) is a computer (EX1005 ¶[0160]) that includes, as shown in Figures 1A and 1B (below), a main processor 102 (“one or more processors”) and two memory units—a main memory unit 104 and a cache memory 140 (individually or collectively, the claimed “non-transitory main memory”) (*id.* ¶¶[0161]–[0165]). The main processor 102 is in communication with the main memory unit 104 (*id.* ¶[0163]) and the cache memory 140 (*id.* ¶[0165]). (See also *id.* ¶[0162], FIGS. 1A–B; *supra* Section IX.A.1.a.)

FIG. 1A

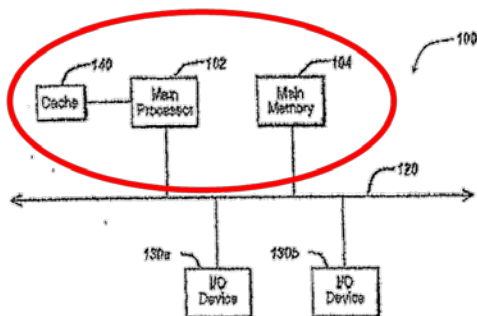
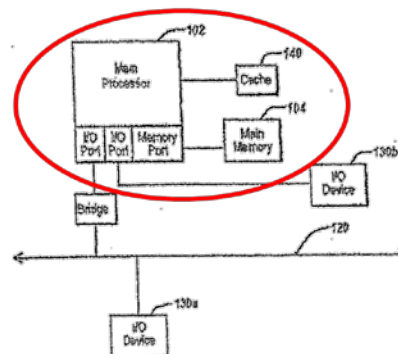


FIG. 1B



(*Id.*, FIGS. 1A–1B (annotated); EX1002 ¶118.)

Therefore, *Wookey* discloses limitation 16.b. (EX1002 ¶119; see also *infra* Sections IX.A.1.c–e.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

- c) **identify idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active;**

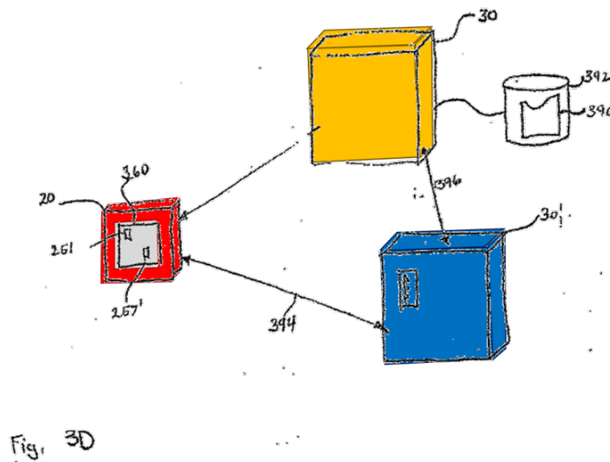
The *Wookey-Eggert* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶120–179.) As explained above, when **client machine 10** uses the code, it results in **client machine 10** operating to initiate a process for establishing a second connection with another **remote machine 30'**. (*Supra* Sections IX.A.1.b.1.) And as discussed below, the process for establishing the second connection includes identifying idle time information, as claimed, in the *Wookey-Eggert* combination. (*Id.*; EX1002 ¶121; *see also infra* Sections IX.A.1.d–f (discussing additional steps in the process for establishing the connection).)

For example, regarding the connection (e.g., connection 394 in Figure 3D) between **client machine 10** and **remote machine 30'**, *Wookey* provides that “the client machine 10 includes an HTTP client agent,” such as a web browser application, which “can use any type of protocol.” (EX1005 ¶[0159]; *see also id.* ¶[0155].) For example, connection 394 can be made using various protocols, including the Virtual Network Computing (VNC) protocol. (*Id.* ¶[0216].) VNC is a desktop-sharing system that allows a computer to remotely access and control another computer. (EX1002 ¶¶122–123; EX1022 ¶[0019].) VNC can run over

Petition for *Inter Partes* Review
 Patent No. 10,075,564

various “industry standard transport protocols, such as TCP/IP, IPX/SPX, NetBEUI.” (EX1005 ¶[0225]; *see also id.* ¶¶[0155], [0215]–[0216]; EX1002 ¶124.)

Wookey further explains that **client machine 10** may use “a different type of protocol than the one used to send the request to the remote machine 30.” (*Id.* ¶[0732]; *id.* ¶¶[0738]–[0739], [0772].)



(*Id.*, FIG. 3D (annotated); EX1002 ¶125.)

Wookey also identifies desired characteristics associated with the second connection that a POSITA would have considered when choosing a protocol such as negotiating parameters related to the connection, reducing unintentional termination of sessions due to an imperfect connection, detecting and handling disconnections like when a mobile device enters an elevator, and ability to specify inactive time prior to connection termination. (EX1002 ¶¶126–127; EX1005 ¶¶[0581], [0721],

Petition for *Inter Partes* Review
Patent No. 10,075,564

[0751], [1134]–[1136], [1153]; *see also id.* ¶¶[0738]–[0739] (disclosing that **remote machine 30'** determines which protocol is being used based on received connection request).)

Wookey further explains that machines **10** and **30'** “exchange a set of messages which negotiate the parameters under which communications will occur.” (EX1005 ¶[0774].) Once negotiations are complete, **client machine 10** and **remote machine 30'**, via the network applications on the machines, can “communicate as necessary” (*id.*) (*Id.* ¶¶[0744], [0773].) Thus, **client machine 10** and **remote machine 30'** both send and receive packets when establishing a connection. (EX1002 ¶128.)

Moreover, when **client machine 10** uses one of the encoded URLs (e.g., the one pointing to **remote machine 30'**) on the HTML page 288, it results in **client machine 10** establishing a connection (e.g., connection 394) with **remote machine 30'** using a transport protocol. (*Id.* ¶129.)

While *Wookey* discloses that various transport protocols (e.g., TCP/IP, IPX/SPX) may be used to establish the second connection and desired characteristics of such a connection (EX1005 ¶[0225]), *Wookey* does not expressly provide details regarding establishing and maintaining the second connection or the types of negotiable parameters used to establish the second connection. (EX1002 ¶130.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

However, it would have been obvious to a POSITA to consider and implement a protocol (which may be “a different type of protocol than the one used to send the request to the remote machine 30” (EX1005 ¶[0732]) that would have met at least some of *Wookey*’s desired characteristics (e.g., such as those identified above) for the second connection between **client machine 10** and **remote machine 30**’ and would have resulted in the code, when used by **client machine 10**, resulting in **client machine 10** identifying “idle information for detecting an idle time period...” like that recited in limitation 16.c. (EX1002 ¶130.)

Eggert describes such a protocol—a TCP-variant protocol using TCP-variant packets—that allows nodes to negotiate or modify timeout timers on a per-connection basis to provide a reliable and efficient network connection. Accordingly, as discussed below, based on the teachings of *Eggert*, the knowledge of a POSITA, and guidance from *Wookey*, it would have been obvious to configure *Wookey*’s **client machine 10** such that when the “code” (e.g., the encoded URLs in the HTML page 288 or the HTML page 288 including the URLs) is used, **client machine 10** uses the browser application 280 to operate per a protocol such as the one described by *Eggert* to initiate establishment of the connection between **client machine 10** and **remote machine 30**’, to “identify idle information for detecting an idle time period, during which, no packet is communicated in a transmission control

protocol (TCP)-variant connection to keep the TCP-variant connection active.” (EX1002 ¶131.)

(1) *Eggert*

Eggert, like *Wookey*, relates to establishing a reliable connection between two nodes. (EX1002 ¶¶132–134.) For instance, where a node may be mobile, the node may experience “change[s] [in] network attachment points based on [its] current location”, and thus *Eggert* discloses networking features in the same technical field as *Wookey*. (EX1006, 1–2; *infra* Section IX.A.1.c.2 (discussing similarities between *Wookey* and *Eggert*).)

Eggert describes a modification to the TCP protocol to “allow established TCP connections to survive periods of disconnection” (EX1006, Abstract.) In particular, *Eggert* introduces a “TCP Abort Timeout Option” (ATO) that “allows conforming TCP implementations to negotiate individual, per-connection abort timeouts.” (*Id.*, Abstract; *see also id.*, 3.) Thus, *Eggert* discloses a TCP-variant protocol that allows established TCP connections to survive periods of disconnection by negotiating abort timeouts. (*Id.*, Abstract; EX1002 ¶135; *see also id.* ¶¶84–93 (*Eggert* overview).)

Eggert explains that its TCP-variant protocol solves the problem “where hosts are only intermittently connected to the Internet.” (EX1006, 1–3.) For example, a

Petition for *Inter Partes* Review
Patent No. 10,075,564

mobile host may “experience disconnected periods during which no network service is available” when “mobile hosts ... change network attachment points.” (*Id.*) These disconnected periods may lead to the “established TCP connections” being “abort[ed] during periods of disconnection.” (*Id.*) *Eggert* overcomes this problem in the art by allowing hosts to negotiate per-connection abort timeouts. (*Id.*, 3.) *Eggert*’s TCP-variant protocol “allow[s] mobile hosts to maintain TCP connections across disconnected periods that are longer than their system’s default abort timeout.” (*Id.*, EX1002 ¶¶136–137.)

Eggert further explains that if “[a] TCP implementation” on a device “does not support the TCP Abort Timeout Option,” then the device “SHOULD silently ignore it.” (EX1006, 7.) *Eggert*, thereby, provides a new TCP option to the traditional TCP implementation where devices configured according to *Eggert* vary from the traditional TCP implementation in terms of the ATO negotiation. (EX1002 ¶¶138–139.) And like the ’564 patent, *Eggert*’s ATO is also incorporated in the TCP header and arranged in a similar format. (*Compare* EX1001, 15:52–60, FIG. 8 with EX1006, 3–5, FIG. 1.)

Petition for *Inter Partes* Review
 Patent No. 10,075,564

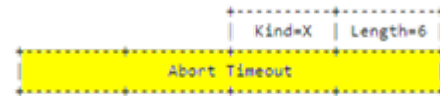
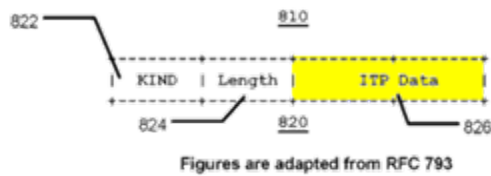


Figure 1: TCP Abort Timeout Option

(EX1001, FIG. 8 (annotated) (left); EX1006, FIG. 1 (annotated) (right); EX1002 ¶140.)

Thus, a connection using *Eggert's* ATO would be a TCP-variant connection, and at least the segments using the TCP ATO for establishing such a TCP-variant connection would be TCP-variant packets. (EX1002 ¶141; *supra* Section VIII; *infra* Section IX.A.1.d.)

Eggert's TCP-variant protocol uses the three-way handshake to establish a connection between two nodes on a network. (EX1006, 5; EX1002 ¶142.) A three-way handshake involves the exchange of synchronize (SYN) and acknowledge (ACK) messages between the nodes to create a connection—i.e., SYN, SYN-ACK, and ACK message in a standard TCP implementation. (EX1002 ¶142; EX1008, 31–32, 34–37, *see also* EX1001, 14:57–15:3 ('564 patent disclosing the standard TCP three-way handshake).)

Petition for *Inter Partes* Review
 Patent No. 10,075,564

As discussed below in Sections IX.A.1.c.1.a–b and exemplified in Figure 2 below, *Eggert* discloses the features of this limitation in two different ways.¹⁵

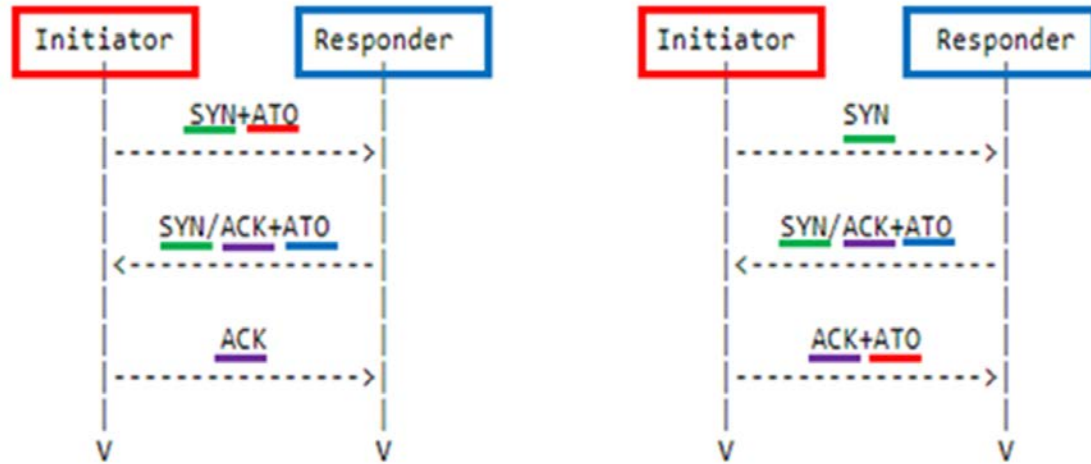


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

(EX1006, FIG. 2 (annotated); EX1002 ¶143.)

In the first way, *Eggert* discloses this feature because when an **initiator** (“client node”) initiates an abort timeout negotiation, the **initiator** must identify the proposed abort timeout value (“idle information”), to include in a “**SYN+ATO**”

¹⁵ In the combined *Wookey-Eggert* process/system discussed below (e.g., *infra* Section IX.A.1.c.2), the combined process/system discloses the claim limitations (e.g., limitations 16.d–f) under both embodiments shown in *Eggert* (i.e., left and right side of Figure 2).

Petition for *Inter Partes* Review
Patent No. 10,075,564

segment, for detecting an abort timeout period (“idle time period”), as explained further below in Section IX.A.1.c.1.a. In the second way, *Eggert* discloses this feature because when the **responder** initiates an abort timeout negotiation, the **initiator** must identify the proposed abort timeout value (“idle information”) in the received “SYN/ACK+ATO” segment, for detecting an abort timeout period (“idle time period”), as explained further below in Section IX.A.1.c.1.b. Under both circumstances, in the combined *Wookey-Eggert* combination, when there is a disconnection preventing communication between the two nodes as envisioned by both *Wookey* and *Eggert*, the combination also discloses this feature, as explained further below in Section IX.A.1.c.2. (EX1002 ¶144.)

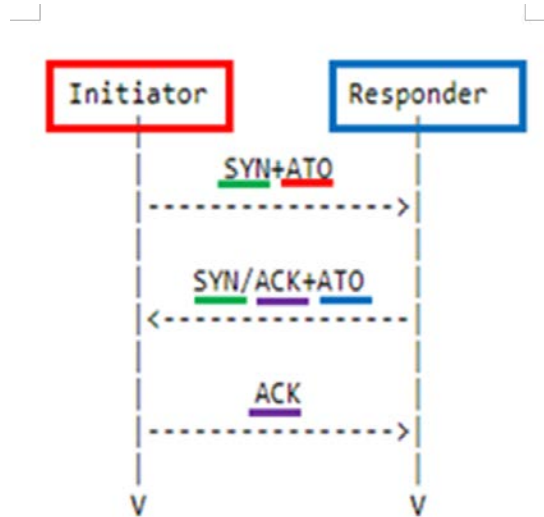
(a) “SYN+ATO” segment

First, regarding the left-hand side of Figure 2 (when the **initiator** initiates an abort timeout negotiation), *Eggert* teaches that the **initiator** (e.g., “client node”) sends the **responder** (e.g., “server node”) an “SYN+ATO” segment.¹⁶ (EX1006,

¹⁶ The ’564 patent describes a term “packet” in a manner consistent with what the TCP standard (RFC 793) describes as a “segment” at the time. (*Compare* EX1001, 14:28–31, FIG.8 *with* EX1008, 15, FIG. 3 (showing the same TCP header in a TCP

Petition for *Inter Partes* Review
 Patent No. 10,075,564

FIG. 2; *see also id.*, 5–7.) The ATO abbreviation in the segment identifies that this segment contains an Abort Timeout Option. (*Id.*, 3.)



(*Id.*, FIG. 2 (cropped and annotated); EX1002 ¶145.)

The TCP header of the SYN segment contains *Eggert's* ATO. (EX1006, 3–5, FIG. 1.) As shown below, the ATO formats use a “Kind” sub-field to indicate the new option, a “Length” sub-field, and an “Abort Timeout” sub-field containing the timeout value. (*Id.*, FIG. 1.)

“packet” (EX1001) and TCP “segment” (EX1008)).) Thus, a POSITA would have understood that *Eggert's* TCP “segment” is the same as the TCP “packet” as described and claimed in the '564 patent. (EX1002 ¶146.)

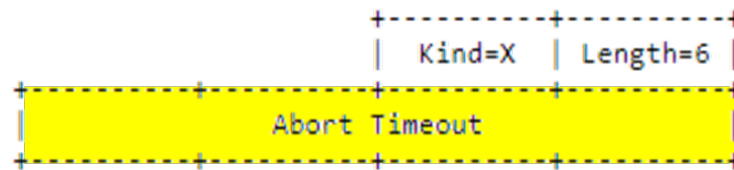


Figure 1: TCP Abort Timeout Option

(*Id.*, FIG. 1 (annotated); EX1002 ¶147.) The **Abort Timeout** sub-field in the ATO conveys an abort timeout value (“idle information”) for detecting an abort timeout period (“idle time period”) between hosts. (*See also* EX1006, FIG. 1; EX1002 ¶147.)

So when the **initiator** initiates an abort timeout negotiation, the **initiator** must identify the proposed abort timeout value to include in the “**Abort Timeout**” sub-field in the ATO. (EX1006, 5, 9.) Therefore, the **initiator** necessarily identifies the proposed abort timeout value (“idle information”) *before* sending a “SYN+ATO” segment. (EX1002 ¶148.) For example, the **initiator** may identify the abort timeout value in the form of a default timeout (EX1006, 3, 7, 9) or as a bound for the negotiated abort timeout (*id.*, 11). These values would be data values stored in memory, which are available to the **initiator** when generating the “SYN+ATO” segment. (EX1002 ¶149; *see also id.* ¶¶150–151.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

The manner in which *Eggert* discloses identifying such values is consistent with the '564 patent, which explains that the idle information may be identified from memory. (EX1001, 10:64–11:4; EX1002 ¶152.)

Eggert further discloses that the abort timeout value (“idle information”) corresponds to an “idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active,” as claimed. (EX1002 ¶153.) For example, *Eggert* explains that the nodes use the abort timeout value to track periods during which no packet is communicated to keep the connection active. (EX1006, 1–7; *see also id.*, 9.) A POSITA would have recognized that the packet that must be “communicated ... to keep the ... connection active” in *Eggert* is an ACK that a node receives in response to a previous sent segment. (EX1002 ¶154.) Indeed, *Eggert*’s ATO relies upon “[t]he TCP specification [1] [which] includes a ‘user timeout’ that defines the maximum amount of time that segments may remain *unacknowledged* before TCP will abort the connection. If a disconnection lasts longer than the user timeout, the TCP connection will abort.” (EX1006, 3.) Like *Eggert*, the '564 patent also describes modifying a TCP user timeout. (*Compare* EX1001, 12:48–51, 21:12–20 *with* EX1006, 3.) Accordingly, *Eggert*’s description of the abort timeout as the duration of time during which a node can wait to receive an ACK is a period “during

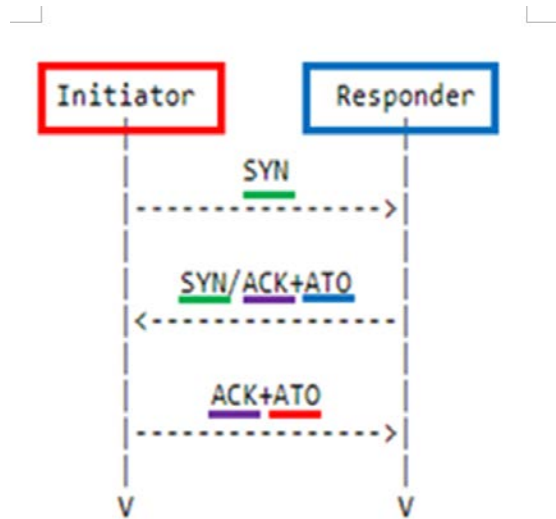
which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active.” (EX1002 ¶155.)

Therefore, *Eggert* discloses the features in this limitation. (*Id.* ¶156.) And as discussed below, it would have been obvious to include such features into *Wookey*’s process/system. (*See infra* Section IX.A.1.c.2.)

(b) “SYN/ACK+ATO”

Regarding the right-hand side of Figure 2 (when the **responder** initiates an abort timeout negotiation), *Eggert* teaches that the **initiator** (“client node”) receives a “SYN/ACK+ATO” segment from a **responder** (“server node”). (EX1006, 5–7.) As explained, the ATO identifies that this segment contains an Abort Timeout Option with a proposed abort timeout for the connection. (*Id.*, 3, 5.) Because the **initiator** is “the receiving host,” the **initiator** must “decide[] whether to accept, shorten, or reject its peer’s proposed abort timeout.” (*Id.*, 5; *see also id.*, 9.) Thus, the **initiator** identifies the proposed abort timeout value (“idle information”) for detecting an abort timeout period (“idle time period”) in the received “SYN/ACK+ATO” segment to accept, shorten, or reject this value for the negotiated timeout value.

Petition for *Inter Partes* Review
 Patent No. 10,075,564



(EX1006, FIG. 2 (annotated); EX1002 ¶¶157–158.)

The manner in which *Eggert* identifies such values is consistent with the '564 patent, which provides that a node can identify idle information from “a message received via a network” (EX1001, 10:55–63) or “identified in a packet [with an ITP header] in the TCP connection” (*id.*, 11:26–28). (EX1002 ¶159.)

Like above for the “SYN+ATO” segment, the abort timeout value in the received “SYN/ACK+ATO” segment corresponds to an idle time period, “during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active,” as claimed. (*Supra* Section IX.A.1.c.1.a; EX1002 ¶160.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

Therefore, *Eggert* discloses the features in this limitation. (EX1002 ¶161.) And as discussed below, it would have been obvious to include such features into *Wookey*'s process/system. (*See infra* Section IX.A.1.c.2.)

(2) Reasons to Combine

Based on *Wookey*'s and *Eggert*'s disclosures, the POSITA's knowledge at the time, and the discussions above, it would have been obvious to configure and implement *Wookey*'s **client machine 10** to use a TCP-variant protocol like the one disclosed by *Eggert* when establishing (and using) the second connection between **client machine 10** and **remote machine 30'** concerning claim limitations 1.c–1.f and claims 22–23. Such an implementation would have had many advantages and been a predictable and straightforward result achieved by combining well-known technologies using known methods. (EX1002 ¶162.) *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 416–18 (2007).

A POSITA would have recognized that *Eggert* and *Wookey* disclose features in a similar technological field. (EX1002 ¶163.) For example, both *Wookey* and *Eggert* are concerned about maintaining connections in situations where at least one node (e.g., a mobile node) in the connection may lose connectivity due to its movement. (*Id.*) *Wookey* discloses that “the client machine 10 may traverse network segments or network access points that cause changes in the network address ... or

Petition for *Inter Partes* Review
Patent No. 10,075,564

causes the client machine 10 to disconnect.” (EX1005 ¶¶[0581]; *see also id.*, ¶¶[1135]–[1136] (discussing the need for connections to manage unintentional disconnection between the client and remote machines), [1153] (discussing an unintentional disconnection when the **client machine 10** enters an elevator).) Similarly, *Eggert* discloses that “[l]engthening abort timeouts allows established TCP connections to survive periods of disconnection” as “[s]ome hosts are only intermittently connected to the Internet.” (EX1006, 1–2.) Like *Wookey*, *Eggert* discloses that “[o]ne example [of such unintentional disconnections] is mobile hosts that change network attachment points based on current location” and “[i]n between connected periods, mobile hosts may experience disconnected periods during which no network service is available.” (*Id.*, 2–3.) Thus, both *Wookey* and *Eggert* are directed to establishing a reliable connection between two nodes, where the two nodes may be mobile nodes that experience an unintentional disconnection, so a POSITA would have had reason to consider *Eggert* when contemplating and implementing the teachings of *Wookey*. (EX1002 ¶¶164–166.)

Moreover, as explained, *Wookey* describes providing access to code that, when used by **client machine 10**, causes **client machine 10** to establish a second connection between **client machine 10** and **remote machine 30'** using a protocol that can be different from the one used to connect **client machine 10** and **remote**

Petition for *Inter Partes* Review
Patent No. 10,075,564

machine 30. *Wookey* further discloses the types of characteristics a POSITA would have considered concerning the protocol that can be used for the second connection, such as negotiating parameters related to the connection, reducing unintentional termination of sessions due to an imperfect connection, detecting and handling disconnections like when a mobile device enters an elevator, and ability to specify a permissible inactive time before connection termination. (EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153]; *see supra* Section IX.A.1.c.) A POSITA would have thus been motivated by *Wookey*’s disclosures and suggestions to consider protocols that would provide the desired features to *Wookey* for the second connection with **remote machine 30’**, including the features associated with the TCP-variant connection described by *Eggert*. Such a POSITA would have been motivated to incorporate a protocol like *Eggert*’s TCP-variant protocol in the use of the “code” by **client machine 10** for establishing a connection between the **client machine 10** and **remote machine 30’**. (EX1002 ¶¶167–168.)

A POSITA would have also recognized the benefits of such a combination because using the TCP variant protocol with an ATO option as described in *Eggert* would have provided **client machine 10** with an improved, reliable connection to **remote machine 30’** with the versatility of negotiated timeout parameters consistent with *Wookey*’s desired characteristics for the second connection. *See KSR*, 550 U.S.

Petition for *Inter Partes* Review
Patent No. 10,075,564

at 416. Indeed, by using a TCP variant protocol with an ATO option (or similar feature as described by *Eggert*), *Wookey*'s method could "negotiate individual, per-connection abort timeouts ... to survive periods of disconnection" (EX1006, 1) over a reliable connection. (EX1002 ¶169.) Thus, *Wookey*'s method for "providing a client with a reliable connection to a host service" (EX1005 ¶[0077]) would have been improved by allowing "[l]ong abort timeout values" so that the "hosts ... tolerate extended periods of [temporary] disconnection," as provided by *Eggert* (EX1006, 5). (EX1002 ¶169.)

Moreover, *Wookey* describes using various reliable communication protocols to establish a second connection. (EX1005 ¶[0216].) As such, a POSITA would have recognized that configuring *Wookey* to utilize a TCP-variant protocol based on *Eggert*'s teachings would have been both a predictable and straightforward implementation. (EX1002 ¶170); see *KSR*, 550 U.S. at 416.

A POSITA would have further recognized that *Eggert*'s protocol aims to avoid or mitigate interruptions caused by intermittent disconnections between the nodes, a problem recognized by *Wookey*. (EX1002 ¶172; EX1006, 1–3; EX1005 ¶[0903].) As such, a POSITA would have been motivated to consider *Eggert*'s features when considering how to implement *Wookey*'s process/system and modify *Wookey* as discussed. Indeed, a POSITA would have appreciated that *Eggert*'s TCP

Petition for *Inter Partes* Review
Patent No. 10,075,564

variant protocol, which allows the nodes to negotiate an abort timeout, would have improved *Wookey*'s process just as described in *Eggert*. (EX1002 ¶172.) Accordingly, a POSITA would have understood that such a combination would have yielded the predictable result of providing a network communication exchange that minimizes disruptions caused by timeout issues. (*Id.* ¶172); see *KSR*, 550 U.S. at 417.

A POSITA would have further recognized that the *Wookey-Eggert* combination would have involved the use of known technologies (e.g., aspects of similar protocols) and design concepts and processes to obtain the foreseeable result of a reliable connection between *Wookey*'s **client machine 10** and **remote machine 30'**. (EX1002 ¶173.). Indeed, the above-modification would have involved substituting features from one reliable protocol from a finite number of available alternative reliable communication protocols, such as that described by *Eggert*. (*Id.*) Thus, a POSITA would have had a reasonable expectation of success in the above-modification. (*Id.*) See *Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1364 (Fed. Cir.

Petition for *Inter Partes* Review
Patent No. 10,075,564

2007) (“only a reasonable expectation of success, not a guarantee, is needed” in an obviousness analysis).¹⁷

A POSITA would thus have found configuring *Wookey*’s system/process such that the “code” (e.g., encoded URLs in HTML page 288 or page 288 including the encoded URLs), when used, would cause the **client machine 10** to utilize a TCP-variant protocol like *Eggert*’s TCP-variant protocol a foreseeable and straightforward implementation. (EX1002 ¶171); see *KSR*, 550 U.S. at 416. For example, as noted above, *Wookey* discloses that “*the URL includes a file, or a reference to a file, that contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (EX1005 ¶[0216].) In light of the reasons discussed above, a POSITA would have been motivated to configure *Wookey*’s encoded URL such that it includes information resulting in **client machine 10** establishing a connection with **remote machine 30’** using the TCP-variant protocol (like that described by *Eggert*). (EX1002 ¶171.) A POSITA

¹⁷ There is no evidence of nonobviousness at this time that would overcome the overwhelming evidence of obviousness as presented herein. Petitioner reserves the right to address any such evidence/argument should PO later present any in this proceeding.

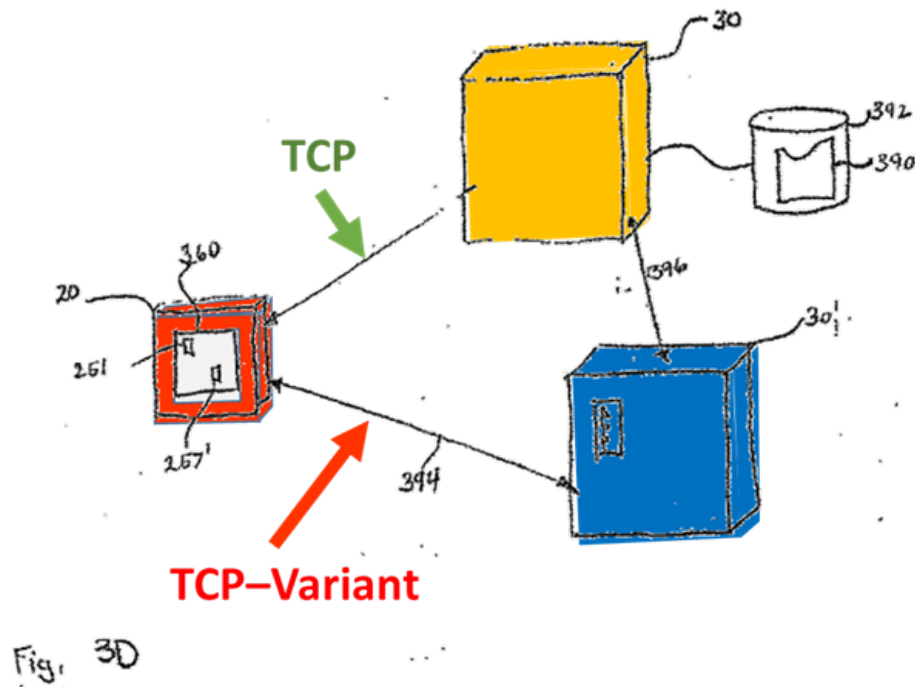
Petition for *Inter Partes* Review
Patent No. 10,075,564

could have achieved this implementation by combining well-known technologies using known methods, such as known network design concepts and technologies described by *Wookey* and *Eggert* and known in the art at the time. (*Id.*)

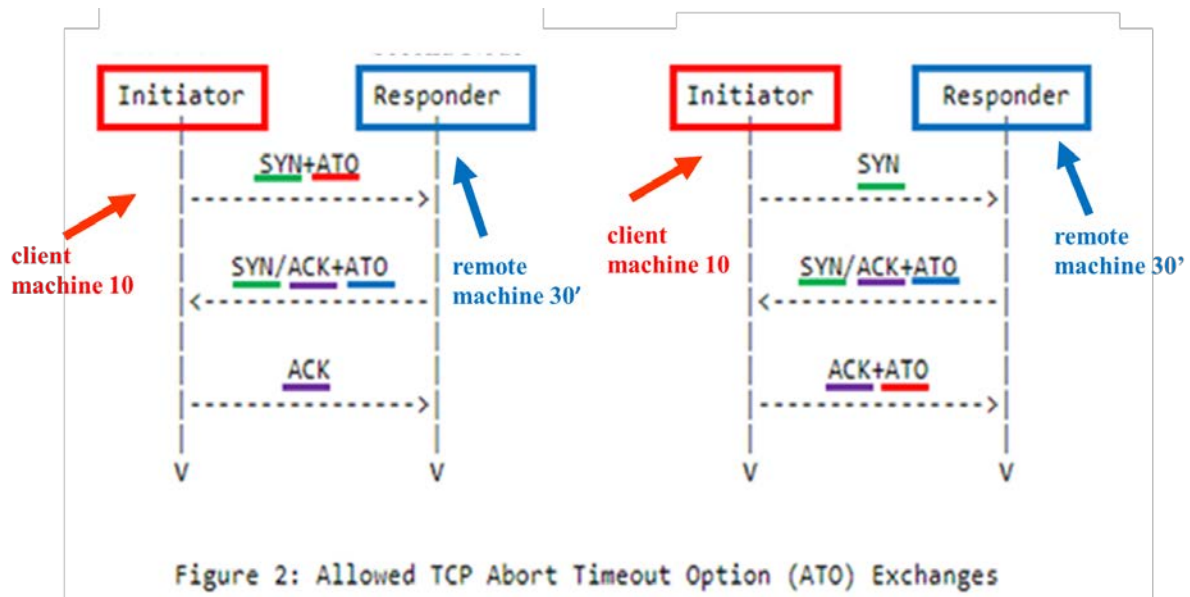
A POSITA would have been skilled and would have had the knowledge to configure *Wookey*'s system/process to implement a TCP-variant protocol in various ways while taking into account any known programming, design, and other related concepts, limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. For example, a POSITA would have been motivated based on such disclosures to configure *Wookey*'s system and process consistent with the exemplary annotated figures from *Wookey* and *Eggert* below.¹⁸

¹⁸ The configurations discussed below are exemplary and not limiting. (EX1002 ¶175.)

Petition for *Inter Partes* Review
 Patent No. 10,075,564



(EX1005, FIG. 3D (annotated); EX1002 ¶174.)



(EX1006, FIG. 2 (annotated); EX1002 ¶174.)

Disconnection

Turning back to the features of limitation 16.c, in addition to the two ways explained above, the features of this limitation are **also** disclosed and/or suggested by the *Wookey-Eggert* combination where a disconnection occurs between **client machine 10** and **remote machine 30**.¹⁹ (EX1005 ¶¶[0581], [1135]-[1136], [1153]; EX1006, 1–3; EX1002 ¶176.)

For example, when an interruption in communication occurs such as a disconnection in the network service (as contemplated by *Wookey* and *Eggert*) or in a physical network medium (as described in the '564 patent) (EX1001, 2:4–8, 2:23–25, 12:61–67), packets from one node do not reach the other node in either direction. (EX1002 ¶177.) In this situation contemplated by the '564 patent, no packets are communicated between the nodes to keep the connection active because they cannot get through the interruption or the physical medium's disconnection. (*Id.*; EX1001, 2:23–25, 12:61–67, FIG. 7 (annotated below in yellow showing no communication

¹⁹ This is consistent with how the '564 patent describes such features. (EX1001, 11:53–62, 2:17–20, 21:11–18 (describing that the idle time period may be linked to the physical disconnection of a network medium or simply a dead connection).)

Petition for *Inter Partes* Review
Patent No. 10,075,564

between the nodes during a disconnection).²⁰ Indeed, neither node would have received any packets from the other node during such a disconnection period. Therefore, neither node would have attempted to send a responsive acknowledgment packet during the period. (EX1002 ¶177.) This is the type of situation where, as discussed above, *Eggert* aims to maintain the connection, e.g., maintain the connection even when no packet is communicated during the negotiated abort timeout period, as there would be no received packets to acknowledge. *Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016) (“[C]ombinations

²⁰ The '564 patent repeatedly confirms that in order for a packet to be “communicated . . . to keep the . . . connection active,” a packet sent by one node must be received by the other node. (EX1002, ¶178; EX1001, 10:34–39 (“idle information for detecting a **first idle time period during which no TCP packet** including data in a first data stream **sent** in the TCP connection **by a second node** is *received by the first node*”), 10:39–44 (disclosing that during the idle time period, no TCP packet “**sent** in the TCP connection **by a second node** is *received by the first node*”), 18:61–19:65 (disclosing that during the idle time period, “**no TCP packet in the TCP connection is *received***, by the first node 602 . . . from second node 604”), FIG. 2 (block 202), FIG. 3 (block 304), FIGS. 6–7.)

Petition for *Inter Partes* Review
 Patent No. 10,075,564

of prior art that sometimes meet the claim elements are sufficient to show obviousness.”); *Power Integrations, Inc. v. Fairchild Semiconductor Int'l, Inc.*, 843 F.3d 1315, 1336-37 (Fed. Cir. 2016) (same).

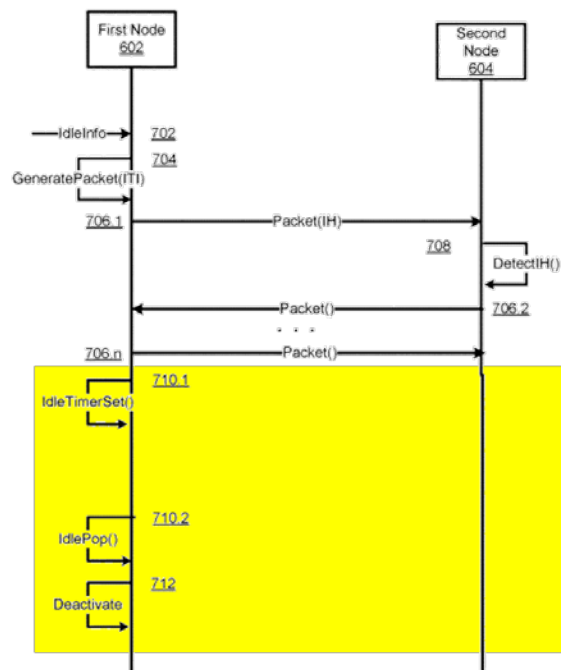


Fig. 7

(EX1001, FIG. 7 (annotated); EX1002 ¶177.)

* * *

Accordingly, the *Wookey-Eggert* combination as discussed above discloses and/or suggests “code” that when used allows the **client machine 10** to identify an abort timeout value (“idle information”) for detecting an abort timeout (“idle time period”) during which no packet is communicated in a TCP-variant connection (e.g.,

Petition for *Inter Partes* Review
Patent No. 10,075,564

Eggert's protocol as implemented in the *Wookey* process/system) to keep that connection active, as claimed in limitation 16.c. (EX1002 ¶179.)

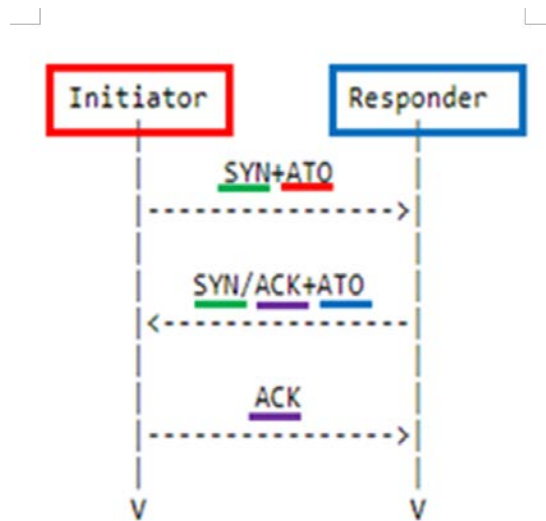
- d) **generate a TCP-variant packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information; and**

The *Wookey-Eggert* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶180–193.) As explained below, in the *Wookey-Eggert* combined process/system, *Eggert* discloses at least two scenarios where the **initiator** (“client node”) performs steps to generate a segment (“TCP-variant packet”) that includes an abort timeout field (“idle time period parameter field”) identifying an abort timeout value (“metadata”) for the abort timeout (“idle time period”) for the connection based on the identified abort timeout value (“idle information”). (*Id.* ¶181; *supra* Section IX.A.1.c; EX1006, 3, 5–7, FIGS. 1, 2.)

(1) **“SYN+ATO”**

Regarding the left-hand side of Figure 2, *Eggert* teaches that the **initiator** generates an “**SYN+ATO**” segment (“a TCP-variant packet”) that contains a TCP Abort Timeout Option (**ATO**) according to an established format to send to the **responder** during a three-way handshake. (EX1006, FIG. 2; *see also id.*, 5–7.)

Petition for *Inter Partes* Review
 Patent No. 10,075,564



(*Id.*, FIG. 2 (annotated); EX1002 ¶182.)

The ATO, as discussed above, incorporated in the generated “SYN+ATO” segment (“TCP-variant packet”) comprises three-parameter fields identifying metadata for the idle time period, as shown in Figure 1 (below): a Kind, Length, **Abort Timeout** field. (See EX1006, 3, FIG. 1; *supra* Section IX.A.1.c.1.) For example, the **Abort Timeout** field of the ATO contains the proposed abort timeout value (“metadata”), represented by a “32-bit value”, for the abort timeout (“idle time period”), specified in seconds, based on the identified abort timeout value (“idle information”) the **initiator** receives from memory accessible to the **initiator**. (EX1006, 3–5; *see also id.*, 9; *supra* Section IX.A.1.c.1.) The other fields—KIND, LENGTH—of the ATO contain data (i.e., bits) each also constitutes “metadata” for

Petition for *Inter Partes* Review
Patent No. 10,075,564

the proposed abort timeout (“idle time period”) that are based on the identified abort timeout value (“idle information”). (EX1002 ¶183.)

The manner in how *Eggert* discloses identifying such values is consistent with the ’564 patent, which explains that the metadata can be a “duration of time” (EX1001, 20:58–63). This is what *Eggert*’s ATO value contains. (EX1006, 3–5, FIG. 1; EX1002 ¶184.) Moreover, the understanding that *Eggert*’s descriptions regarding identifying values, such as the abort timeout value, discloses identifying metadata like that claimed is consistent with PO’s view of the same claimed features. (EX1017, 49 (PO alleging “identifying metadata (e.g., *a value*, etc.) for the idle time period”).)

Also, the “SYN+ATO” segment is a TCP-variant packet since it uses the same format as outlined in the TCP standard but adds the ATO field. (EX1002 ¶185; *supra* Section IX.A.1.c.1.) Indeed, *Eggert*’s TCP ATO shown in Figure 1 uses the same TCP structure as the ITP field illustrated in Figure 8 of the ’564 patent. For example, like the ’564 patent, *Eggert*’s ATO is also incorporated in the TCP header and arranged in a similar format. (*Compare* EX1001, 15:52–60, FIG. 8 with EX1006, 3–5, FIG. 1.) As shown below, both formats use a KIND sub-field to indicate the new option, a TCP option length sub-field, and a **data portion** of the TCP

Petition for *Inter Partes* Review
 Patent No. 10,075,564

option containing the timeout value. (*Compare* EX1001, FIG. 8 with EX1006, FIG. 1.)



(EX1001, FIG. 8 (annotated) (left); EX1006, FIG. 1 (annotated) (right); EX1002 ¶185.) Thus, *Eggert*'s ATO uses the same format as the embodiment disclosed in the '564 patent for a TCP-variant packet. (EX1002 ¶186.)

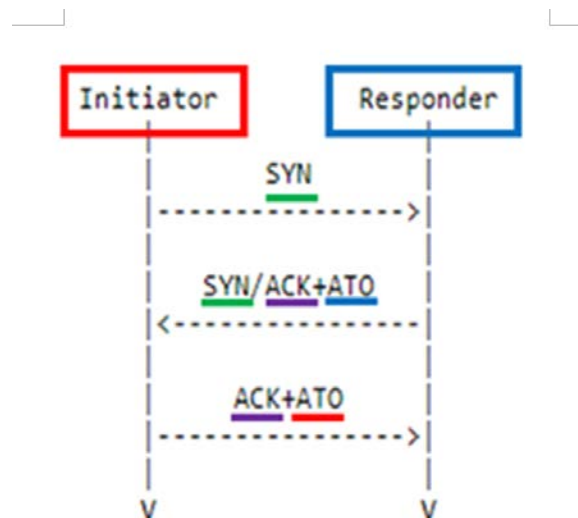
The understanding that *Eggert*'s "SYN/ACK+ATO" segment is a TCP-variant packet in the context of the '564 patent is also consistent with PO's proposed construction of "variant" in a related matter. (*See, e.g.*, EX1018, 60 (Exhibit C) (PO defining a TCP-variant as one that "manifest[s] variety, deviation, or disagreement, [or] var[ies] slightly from the [TCP] standard.")) *Eggert* describes a TCP-variant packet under PO's interpretation because the packets exchanged during the three-way handshake in *Eggert* vary, as described above, from those exchanged in the TCP standard.

As a result, segments generated and sent according to this TCP implementation that uses an ATO option as described in *Eggert* would be TCP-variant packets. (*See also supra* Section VIII.)

Eggert thus discloses the features of this limitation. (EX1002 ¶187.)

(2) “ACK+ATO”

Regarding the right-hand side of Figure 2, *Eggert* teaches that the **initiator** generates an “ACK+ATO” segment, that contains a TCP Abort Time Option (ATO), to send to the **responder** during the three-way handshake. (EX1006, FIG. 2; *see also id.*, 5–7.)



(*Id.*, FIG. 2 (annotated); EX1002 ¶188.)

Like above, the ATO incorporated in the generated “ACK+ATO” segment (“TCP-variant packet”) comprises three fields, the Kind, Length, and **Abort Timeout**

Petition for *Inter Partes* Review
Patent No. 10,075,564

fields, they each constitutes metadata for the abort timeout (“idle time period based on the idle information.”). (EX1006, 3–5, FIG. 1.) For example, the Abort Timeout field of the ATO contains the responsive abort timeout value, represented as a 32-bit value (“metadata”) for the idle time period, specified in seconds, based on the identified proposed abort timeout value (“idle information”) the **initiator** received from the “SYN/ACK+ATO” segment. (*Id.*, 3–5; *see also id.*, 9; *supra* Section IX.A.1.c.1.; EX1002 ¶189.)

The manner as to how *Eggert* discloses identifying such values is consistent with the ’564 patent’s discussions and PO’s view of the same claimed features as discussed above. (*Supra* Section IX.A.1.d.1; EX1002 ¶189.)

The “ACK+ATO” segment is a TCP-variant packet since it uses the same format as outlined in the TCP standard but adds the ATO field. (*Supra* Section IX.A.1.d.1; EX1002 ¶189.) As above, *Eggert*’s ATO uses the same format as the embodiment disclosed in the ’564 patent for a TCP-variant packet. (*Id.*) As a result, segments generated and sent according to this TCP implementation that use an ATO option as described in *Eggert* would be TCP-variant packets. (*See also supra* Section VIII; EX1002 ¶190.)

Thus, *Eggert* discloses the features of limitation in this additional way. (EX1002 ¶191.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

* * *

For reasons similar to those explained above for limitation 16.c, a POSITA would have been motivated to implement in the *Wookey-Eggert* combined process of establishing the connection between **remote machine 30'** and **client machine 10**, the process of generating, by **client machine 10**, a TCP-variant packet (e.g., “SYN+ATO” and/or “ACK+ATO” segment) that includes an **Abort Timeout** field (“idle time period parameter field”) identifying an abort timeout value (“metadata”) for the abort timeout (“idle time period”) for the connection based on the idle information (e.g., the abort timeout value identified before generating “SYN+ATO” segment and/or after receiving the “SYN/ACK+ATO” segment.) (*Supra* Section IX.A.1.c (discussing the *Wookey-Eggert* combination).) Accordingly, the *Wookey-Eggert* combination discloses and/or suggests limitation 16.d. (EX1002 ¶¶192–193.)

- e) **send, from the client node to a server node, the TCP-variant packet to provide the metadata for the idle time period to the server node,**

The *Wookey-Eggert* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶194–200.) As explained below, *Eggert* discloses at least two scenarios where the **initiator** (e.g., “client node”) sends the “SYN+ATO” and “ACK+ATO” segments (“TCP-variant packet”) to the

responder (e.g., “server node”) to provide the abort timeout value (“metadata”) to the **responder** for negotiating an abort timeout for the TCP-variant connection. (EX1002 ¶196; *see also* EX1006, 5–7, FIG. 2.)

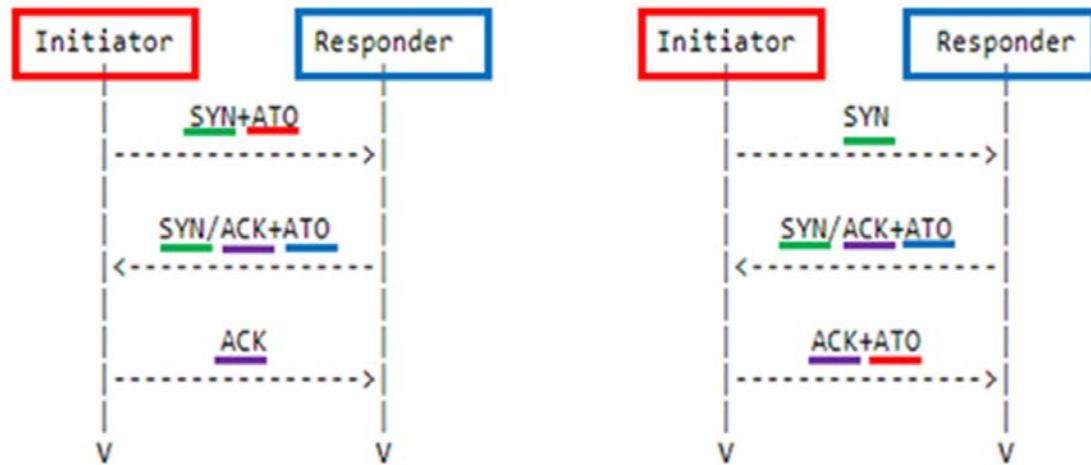


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

(EX1006, FIG. 2 (annotated); EX1002 ¶196.)

(1) “SYN+ATO”

Regarding the left-hand side of Figure 2, *Eggert* teaches that the **initiator** sends to the **responder** a “SYN+ATO” segment (“TCP-variant packet”). (EX1006, FIG. 2; *see also id.*, 5–7; *supra* Section IX.A.1.d.1.) Upon receipt of this segment, the **responder** decides whether to accept, shorten, or reject the **initiator**’s proposed abort timeout. (*Id.*, 5; *see also id.*, 9.) Thus, the **initiator** (“client node”) sends the “SYN+ATO” segment (“TCP-variant packet”) to the **responder** (“server node”) to

provide the proposed abort timeout value (“metadata”) for the abort timeout (“idle time period”) of the connection. (EX1002 ¶197.)

(2) “ACK+ATO”

Regarding the right-hand side of Figure 2, *Eggert* teaches that the **initiator** sends to the **responder** (e.g., “server node”) an “ACK+ATO” segment (“TCP-variant packet”). (EX1006, FIG. 2; *see also id.*, 5–7; *supra* Section IX.A.1.d.2.) And the **responder** (e.g., “server node”), who “initially proposed the Abort Timeout Option” (EX1006, 7) in the “SYN/ACK+ATO” segment, “analyzes the next segment it receives from its peer” (*id.*) to identify the responsive abort timeout value. The responsive abort timeout value specified in the “ACK+ATO” segment is used by both the **initiator** and **responder** for the connection. (*Id.*) Thus, the **initiator** (“client node”) sends the “ACK+ATO” segment (“TCP-variant packet”) to the **responder** (“server node”) to provide the responsive abort timeout (“metadata”) for the abort timeout (“idle time period”) of the connection. (EX1002 ¶198.)

* * *

For reasons similar to those explained above for limitation 16.c, a POSITA would have been motivated to implement in the *Wookey-Eggert* combined process of establishing the connection between **remote machine 30’** and **client machine 10**, the process of sending, by **client machine 10** to **remote machine 30’**, the generated

Petition for *Inter Partes* Review
Patent No. 10,075,564

TCP-variant packet (e.g., “SYN+ATO” and/or “ACK+ATO” segment) to provide a proposed or responsive abort timeout value (“metadata”) for the abort timeout (“idle time period”) of the connection to the **remote machine 30’**. (*Supra* Section IX.A.1.c (discussing the *Wookey-Eggert* combination).) Accordingly, the *Wookey-Eggert* combination discloses and/or suggests limitation 16.e. (EX1002 ¶¶199–200.)

- f) **[the metadata ...], for use by the server node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.**

The *Wookey-Eggert* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶201–213.) As explained below, the *Wookey-Eggert* combination discloses and/or suggests that the abort timeout value (“metadata”) provided by **client machine 10** is for use by **remote machine 30’** (“server node”) to modify a timeout attribute associated with the connection (“TCP-variant connection”) in at least two ways. (EX1002 ¶¶201–203.)

For instance, *Eggert* discloses that after successful negotiation, the nodes use the abort timeout for the connection. (EX1006, 3, 7 (“If the next reply segment does not contain an Abort Timeout Option, the connection MUST use the default abort timeout. If it does, the connection MUST use the abort timeout contained inside the

Abort Timeout Option.”).) Figure 2 reveals the two scenarios for ATO negotiation, both of which disclose limitation 16.f.

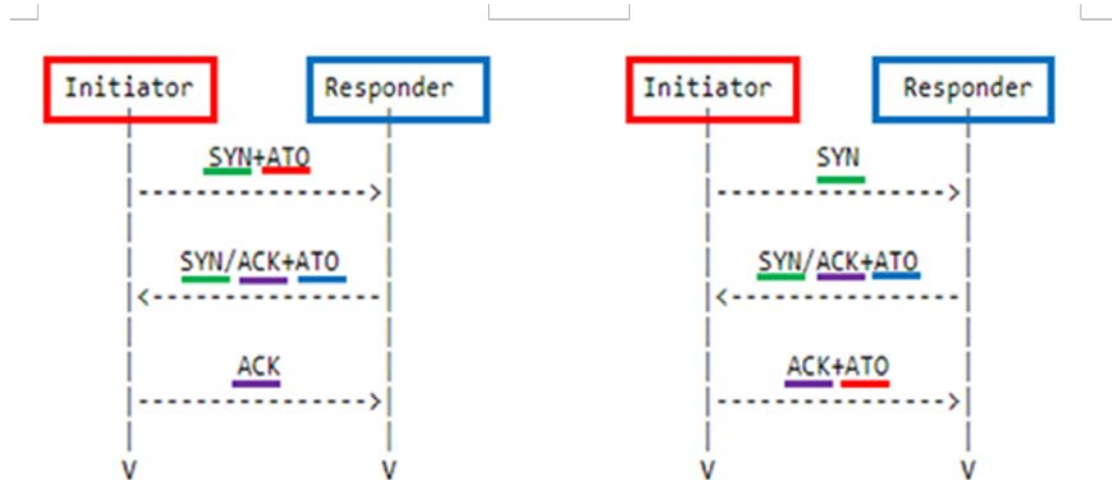


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

(*Id.*, FIG. 2 (annotated); EX1002 ¶204.)

(1) “SYN+ATO”

Regarding the left-hand side of Figure 2, because the **initiator** (e.g., “client node”) proposed the abort timeout value (“metadata”), the **responder** (e.g., “server node”) decides whether to accept, shorten, or reject the proposed abort timeout, resulting in the responsive abort timeout value (“timeout attribute”) for the TCP-variant connection. (EX1006, 5; *see also id.*, 5–7, 9, FIG. 2.) For example, “[t]o shorten the offer,” the **responder** uses the proposed timeout value (“metadata”) as an upper bound on the responsive timeout value (“timeout attribute”) (*id.*, 11)—i.e.,

the **responder** “lowers the timeout value accordingly before sending” (*id.*, 9). (See also *id.*, 5–7.) Thus, the received proposed abort timeout value in the “SYN+ATO” segment is used by the **responder** to modify, based on the proposed abort timeout value, the responsive abort timeout value associated with the TCP-variant connection. (EX1002 ¶¶205–207.)

(2) “ACK+ATO”

Regarding the right-hand side of Figure 2, because the **responder** (e.g., “server node”) initiated an abort timeout negotiation by proposing an abort timeout value, the **responder** must use the responsive abort timeout value (“metadata”) contained inside the ATO to modify its abort timeout (“timeout attribute”)—e.g., by replacing the **responder**’s default abort timeout for this connection. (EX1006, 3, 7.) For example, the **responder**, who initially proposed the abort timeout value in the “SYN/ACK+ATO” segment, analyzes the received “ACK+ATO” for the responsive abort timeout value contained inside the ATO to modify its abort timeout associated with the connection to use the same abort timeout as the **initiator**. (*Id.*, 1–3, 7.) Thus, the received responsive abort timeout value (“metadata”) in the “ACK+ATO” segment (“TCP-variant packet”) is used by the **responder** (“server node”) to modify, based on the responsive abort timeout value (“metadata”), its abort

Petition for *Inter Partes* Review
Patent No. 10,075,564

timeout (“timeout attribute”) associated with the TCP-variant connection.
(EX1002 ¶208.)

* * *

The ’564 patent describes timeout attributes broadly and consistent with *Eggert*’s abort timeout value. (EX1001, 21:13–20 (“ITP option handler component 562 may [*sic*] one or more attribute option handler components 564 to modify one or more corresponding attributes of a keep-alive option, a TCP user timeout, a retransmission timeout, an acknowledgment timeout, and another timeout associated with the TCP connection, in response to identifying the ITP header”), 12:48–52; EX1002 ¶¶209–210.)

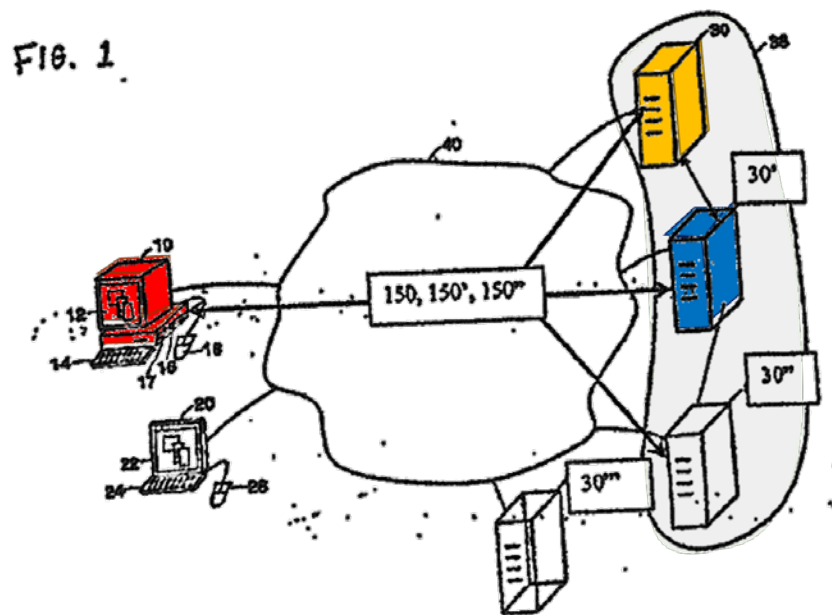
For reasons similar to those explained above for limitation 16.c, a POSITA would have been motivated to implement in the *Wookey-Eggert* combined process of establishing the connection between **remote machine 30’** and **client machine 10**, the feature of **remote machine 30’** using the above-discussed metadata (*see* Sections IX.A.1.e–f) to modify a timeout attribute associated with the TCP-variant connection, like that claimed in limitation 16.f. (*Supra* Section IX.A.1.c (discussing the *Wookey-Eggert* combination).) Accordingly, the *Wookey-Eggert* combination discloses and/or suggests limitation 16.f. (EX1002 ¶¶211–213.)

2. Claim 22

- a) **A system including the non-transitory computer readable medium of claim 16, wherein the system:**

To the extent limiting, the *Wookey-Eggert* combination discloses and/or suggests and renders obvious the features of this preamble.²¹ (*Supra* Section IX.A.1; EX1002 ¶¶214–217.) For example, *Wookey* provides an environment (“system”) (Figure 1) including **client machine 10** (“client node”), **remote machine 30’** (“server node”), and **remote machine 30**. (EX1005 ¶[0020]; *see also id.* ¶¶[0002], [0016], [0196].)

²¹ The analysis above for claim 16, including the reasons for combining *Wookey* and *Eggert*, and the resulting combination, are incorporated and relied upon for claim 22.



(*Id.*, FIG.1 (annotated); EX1002 ¶215.)

As discussed above for claim 16, **remote machine 30** includes the non-transitory computer readable medium (e.g., main memory unit 104 and/or cache memory 140) of claim 16. (*Supra* Sections IX.A.1.a–b; EX1002 ¶¶215–216.)

Thus, the *Wookey-Eggert* combination discloses and/or suggests the features of limitation 22.a. (EX1002 ¶217; *infra* Sections IX.A.2.b–c.)

- b) **includes another server node separate from the server node, is configured to communicate the code to the client node for use in establishing the TCP-variant connection with the server node,**²²

The *Wookey-Eggert* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶218–222.) For example, the *Wookey-Eggert* combination discloses or suggests that the system includes **remote machine 30** (“another server node”), which is separate from **remote machine 30’** (“server node”). (*Supra* Section IX.A.2.a.)

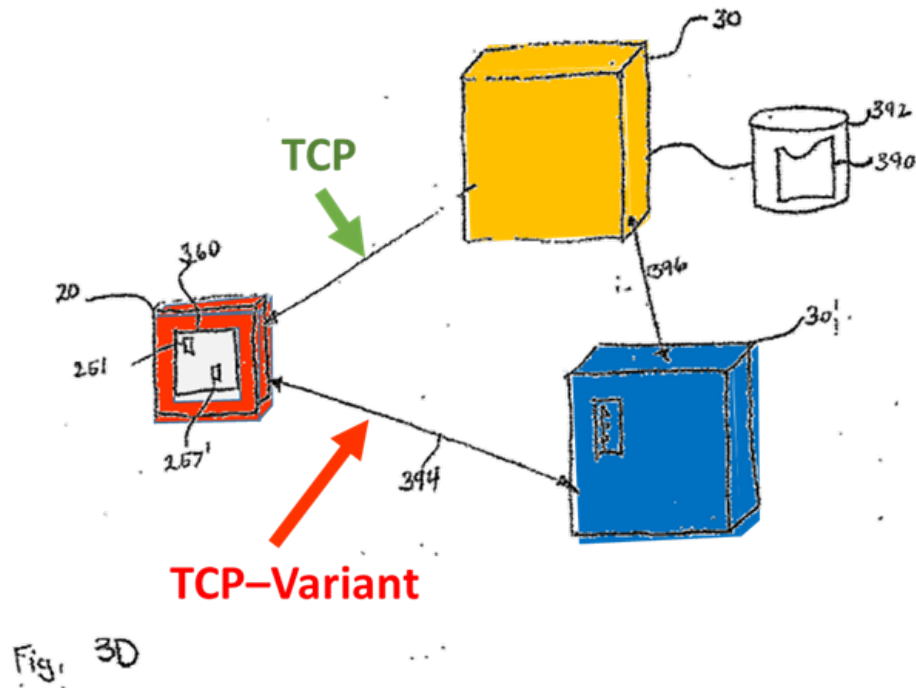
The **remote machine 30** (“another server node”), as explained above, is configured to communicate the code (e.g., the encoded URLs in the HTML page 288 or the HTML page 288 including the encoded URLs) to **client machine 10** for use in establishing the TCP-variant connection (e.g., connection 394 in Figure 3D) with **remote machine 30’** (“server node”). (*Supra* Sections IX.A.1.b–c; EX1002 ¶219.) An exemplary arrangement relating to the above-discussed combination that a POSITA would have considered is illustrated in exemplary fashion below concerning Figure 3D of *Wookey*.²³ Thus, the *Wookey-Eggert* combination discloses and/or suggests and renders obvious this limitation for at least

²² See *supra* n.10. (See also EX1002, ¶218 n.14.)

²³ See n.9.

Petition for *Inter Partes* Review
 Patent No. 10,075,564

the same reasons discussed above for limitations 16.b–c. (*Supra* Sections IX.A.1.b–c.)



(EX1005, FIG. 3D (annotated); EX1002 ¶¶220–222.)

c) **Claim 22.c**

The *Wookey-Eggert* combination discloses and/or suggests and renders obvious the features of limitation 22.c, as explained below in two parts. (EX1002 ¶¶223–226.)

- (1) **and is further configured to perform a 3-way TCP handshake for establishing, with the client node, a TCP connection that is different than the TCP-variant connection []**

The *Wookey-Eggert* combination discloses and/or suggests this limitation. (EX1002 ¶¶227–235.) As discussed above, **remote machine 30** (“another server node”) communicates with **client machine 10** (“client node”) over a TCP connection. (*Supra* Section IX.A.1.b.) As explained below, when communicating over a TCP connection, **remote machine 30** and **client machine 10** would necessarily perform a three-way handshake to set up the TCP connection. (EX1002 ¶228.) Furthermore, this TCP connection between **remote machine 30** (“another server node”) and **client machine 10** (“client node”) is different than the TCP-variant connection (between **client machine 10** and **remote machine 30**). (*Id.* ¶228.)

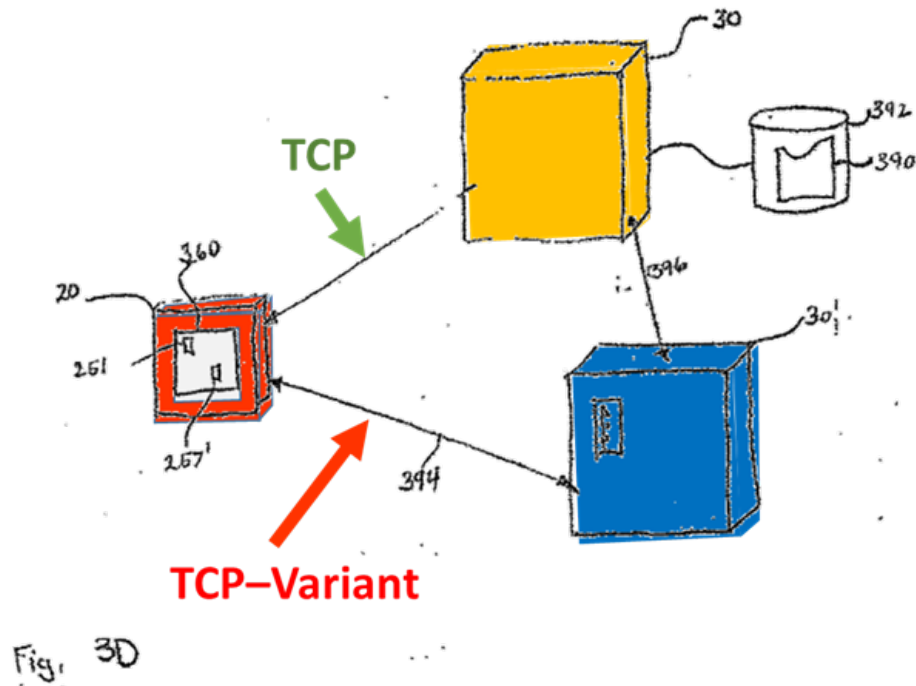
First, a three-way handshake method is necessarily performed to establish a TCP connection. (*Id.* ¶¶229–230.) For example, RFC 793 (EX1008), the TCP specification, provides that a “three way handshake is necessary” (EX1008, 32) and

Petition for *Inter Partes* Review
Patent No. 10,075,564

that it is “the procedure used to establish a [TCP] connection” (*id.*, 34.). (*See also id.*, 31–32, 34–37; EX1002 ¶229.) The TCP three-way handshake involves the transmission of three messages to negotiate and start a TCP-based session between the two devices. (*Id.* ¶229; EX1008, 31–32, 34–37.) The ’564 patent acknowledges that the three-way handshake is required to establish a TCP connection, citing RFC 793. (EX1001, 13:49–63.) Thus, to establish a TCP connection between **client machine 10** and **remote machine 30**, **remote machine 30** is necessarily configured to perform a three-way TCP handshake because this handshake is a *required* procedure for establishing a TCP connection, as claimed. (EX1002 ¶230.)

Second, this TCP connection between **remote machine 30** and **client machine 10** is different from the TCP-variant connection between **remote machine 30’** and **client machine 10** in the *Wookey-Eggert* combination. (*Id.* ¶231; *supra* Section IX.A.1.c.) Indeed, as discussed for limitation 16.c, *Wookey* explains that **client machine 10** may use a different protocol than the one used to send the request to **remote machine 30** when establishing the second connection with **remote machine 30’**. (*Supra* Section IX.A.1.c; EX1005 ¶[0732].) The *Wookey-Eggert* combination discloses and/or suggests that **client machine 10**, as shown in Figure 3D, may establish a connection with **remote machine 30’** that utilizes a *different* protocol (e.g., the TCP-variant protocol described concerning *Eggert*) than the TCP

protocol used with the first connection between **client machine 10** and **remote machine 30**. (EX1002 ¶232.) An exemplary arrangement relating to the above-discussed combination that a POSITA would have considered is exemplified below for Figure 3D of *Wookey*.²⁴



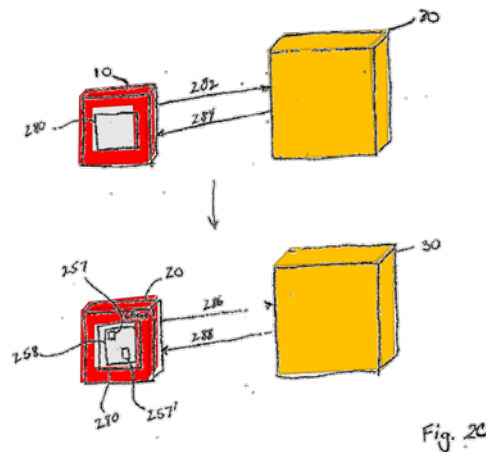
(EX1005, FIG. 3D (annotated); EX1002 ¶¶233–235.)

²⁴ See n.9

- (2) [] for permitting the client node to fetch the code in addition to other data from the system via a hypertext transfer protocol (HTTP).

The *Wookey-Eggert* combination discloses and/or suggests this limitation. (EX1002 ¶¶236–243.) For example, **client machine 10** (“client node”) and **remote machine 30** (“another server node”) in the combined *Wookey-Eggert* system communicate over an HTTP-based web browser application 280 utilizing the TCP connection, to permit **client machine 10** to request and receive (“fetch”) code (e.g., the encoded URLs in the HTML page 288 or the HTML page 288 including the encoded URLs). (*See supra* Sections IX.A.1.b.1–2; *see also supra* Section IX.A.2.c.1.) The TCP connection established between **client machine 10** and **remote machine 30** permits the client machine to receive the code via an HTTP-based application (e.g., the web server and web browser application 280). (EX1002 ¶¶237–238; *see also supra* Section IX.A.1.b.) Figure 2C depicts an arrangement where **client machine 10** uses an HTTP-based web browser application 280 to communicate with a **remote machine 30**. (*E.g.*, EX1005 ¶¶[0021], [0159], [0192]–[0193].)

Petition for *Inter Partes* Review
 Patent No. 10,075,564



(*Id.*, FIG. 2C (annotated); EX1002 ¶¶237–238.)

For example, **remote machine 30** hosts an HTML page. (EX1005 ¶¶[0192]–[0193], FIG. 2C; *see also id.*, ¶¶[0024], [0155].) The **client machine 10** executes the web browser application 280 (*id.* ¶[0192]) that “transmits a request 282 to access a Uniform Resource Locator (URL) address corresponding to an HTML page residing on remote machine [30]” (*id.* ¶[0193]).²⁵ Following this request, **remote machine 30** returns an HTML page to **client machine 10**. (*Id.*) For example, **remote machine 30** may first return “an authentication page that seeks to identify the client machine 10 or the user of the client machine 10.” (*Id.*) After authentication, **remote machine 30** may return “an HTML page 288 that includes a

²⁵ *See* n.13.

Petition for *Inter Partes* Review
Patent No. 10,075,564

Resource Neighborhood window 258 in which appears graphical icons 257, 257' representing resources to which the client machine 10 has access.” (*Id.* ¶[0196].) The HTML pages include data, such as tags and associated information, defining the layout of the webpage and what is to be presented on the webpage. (EX1002 ¶¶239–240; *see also* EX1005 ¶¶[0196], [0207]–[0214], [0754]–[0755].)

Thus, the TCP connection established between **client machine 10** and **remote machine 30** permits a client machine to fetch an HTML page 288 (“code”) and encoded URLs (also “code”), in addition to an authentication page (including “other data”) via an HTTP-based application (the web server and web browser application 280). (EX1002 ¶¶241–243; *see also supra* Section IX.A.1.b.)

3. Claim 23

- a) **The system of claim 22, wherein the code, when used by the client node, results in the client node operating such that the TCP-variant connection is established between the client node and the server node instead of the TCP connection in order to permit communication, between the client node and the server node, of the timeout attribute,**²⁶

The *Wookey-Eggert* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶244–248.) As described above for claim 22, the combined *Wookey-Eggert* system discloses and/or suggests that **remote machine 30** causes the code (e.g., the encoded URLs in the HTML page 288 or the HTML page 288 including the encoded URLs) to be sent to **client machine 10** via a TCP connection. (*Supra* Sections IX.A.1.b, IX.A.2.) This code, when used by **client machine 10**, causes **client machine 10** to communicate with **remote machine 30'**, using the TCP-variant protocol, instead of the TCP protocol, to establish the TCP-variant connection. (*Supra* Sections IX.A.1.b–f, IX.A.2.) During the setup of the TCP-variant connection, in the combined *Wookey-Eggert* system, **client machine 10** and the **remote machine 30'** would have negotiated an abort timeout for the connection. (*Supra* Sections IX.A.1.c–f; EX1002 ¶¶245–246.)

²⁶ See *supra* n.10. (See also EX1002, ¶244 n.15.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

Thus, the code, when used by **client machine 10** (“client node”), results in **client machine 10** operating such that the TCP-variant connection is set up between **client machine 10** and **remote machine 30’** (“server node”) instead of the TCP connection to permit communication, between **client machine 10** and **remote machine 30’**, of an abort timeout value which is used to set up the timeout attribute for the TCP-variant connection (as discussed above from limitation 16.f). (EX1002 ¶¶247–248; *see also supra* Sections IX.A.1.c–f.) So, the abort timeout value used to set up the timeout attribute is not communicated over a TCP connection but instead over the TCP-variant connection. (EX1002 ¶¶247–248.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

- b) **where the timeout attribute is not communicated when establishing the TCP connection between the system and the client node, but is communicated when establishing the TCP-variant connection between the client and the server node.**²⁷

The *Wookey-Eggert* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶249–253.) As explained above, in the *Wookey-Eggert* combination, the timeout attribute—e.g., the abort timeout value for the connection—is a value associated with its TCP-variant connection between **client machine 10** (“client node”) and **remote machine 30’** (“server node”). (*Supra* Sections IX.A.1.b–f, IX.A.2, IX.A.3.a.) The abort timeout value associated with the TCP-variant abort timeout for **remote machine 30’** is communicated when establishing the TCP-variant connection between **remote machine 30’** and **client machine 10** (“client node”), and not when setting up a TCP connection between **remote machine 30** (“another server node”) and **client machine 10**. (*Supra* Sections IX.A.1.b–f, IX.A.2, IX.A.3.a; EX1002 ¶250.)

Additionally, the abort timeout value associated with the timeout attribute is not communicated when setting up the TCP connection between “the system and the

²⁷ Claim 23 does not provide any antecedent basis for “the TCP connection between the system and the client node.” However, Petitioner assumes for this proceeding

Petition for *Inter Partes* Review
Patent No. 10,075,564

client node” (e.g., between **remote machine 30** and **client machine 10**), but rather are communicated via a TCP-variant packet—e.g., “SYN+ATO” segment, SYN/ACK+ATO” segment, and/or “ACK+ATO” segment—when establishing the TCP-variant connection between **client machine 10** and the **remote machine 30**. (*Supra* Sections IX.A.1.c–f, IX.A.2, IX.A.3.a; EX1002 ¶¶251–252.) For these reasons, the *Wookey-Eggert* combination discloses and/or suggests the features of limitation 23.b. (EX1002 ¶253.)

that limitation 23.b was intended to recite “the TCP connection between the another server node and the client node,” based on limitations 22.b–c. Under that assumption, *Wookey-Eggert* combination discloses or suggests this limitation. Petitioner does not concede that claim 23 is not indefinite and reserves the right to challenge the definiteness of this claim in other proceedings.

X. EGGERT IS A PRINTED PUBLICATION

Eggert is an IETF Internet-Draft (“ID”) and is prior art under at least 35 U.S.C. §102(b) (pre-AIA) because it was published in April, 2004. The declaration of Alexa Morris, Managing Director of IETF, confirms *Eggert* was published, disseminated, and reasonably available to the public by April, 2004. (EX1019 ¶¶1–3, 9–10.)

IDs were at the time, and continue to be, working documents published through the IETF Secretariat and disseminated to the public by IETF through various media so that others may comment on them. (*Id.* ¶¶4–6.) Since 1998 (including 2004 and today), the IETF Secretariat publishes an ID on its public website, and publication announcements were sent to an IETF mailing list and relevant working group mailing lists. (*Id.* ¶¶5–6.) Anyone could have subscribed to IETF mailing lists, and the archives of all IETF mailing lists are publicly available on IETF’s website. (*Id.* ¶¶6–8.)

Eggert was published by the IETF in April, 2004. (EX1019 ¶¶9–10.) In 2004, a POSITA could have learned about *Eggert* in various ways, such as through announcements on the IETF announce mailing list, discussions on the IETF tcpm working group mailing list, or review of the archives of the IETF announce or tcpm mailing lists. (*Id.* ¶¶9–10.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

Collectively, this demonstrates that *Eggert* was published and publicly available in 2004 and at least prior to 2010. *Polycom, Inc., v. Directpacket Research, Inc.*, IPR2019-01235, Paper 19, 30–32 (Jan. 13, 2020) (finding a similar IETF ID was publicly available based on similar evidence).

XI. DISCRETIONARY DENIAL IS NOT APPROPRIATE

A. *Eggert* is Not Cumulative

Eggert is not cumulative to any prior art the Examiner considered during prosecution of the '564 patent and no arguments similar to those contained herein were ever presented to or considered by the Office. Nonetheless, PO may assert here (as it did in the Google -845 IPR) that RFC 5482 (EX1016), listed on the face of the '564 patent, is substantially similar to *Eggert*. It is not, as the Board found in the Google -845 IPR. Google -845 IPR, Paper 16 at 14-17. The Board should reach the same conclusion here.

First, the Board routinely institutes trial where references in an IPR were considered in an IDS but not relied upon to reject claims during prosecution. *See, e.g., Amber.IO, Inc. D/B/A Two Tap v. 72Lux, Inc. D/B/A Shoppable*, IPR2020-00015, Paper 8 at 18-20 (April 1, 2020) (declining to exercise discretion where references were cited in an IDS but no evidence that the references were substantively considered) (citing *Advanced Bionics, LLC v. Med-El*

Petition for *Inter Partes* Review
Patent No. 10,075,564

Elektromedizinische Geräte GmbH, IPR2019-01469, Paper 6 (February 13, 2020) (precedential)); *Apple, Inc. v. Omni Medsci, Inc.*, IPR2020-00029, Paper 7 at 52–55 (April 22, 2020) (similar). Consistent with those proceedings, RFC 5482 was cited during prosecution along with twenty other references in an IDS with no explanation regarding any relevant disclosures. (EX1004, 70–72.) Moreover, RFC 5482 was never relied upon by the Examiner to reject the claims. (See EX1004, 27–30.) Even assuming *Eggert* is cumulative to RFC 5482, which it is not (as explained below), *Eggert* was not considered in the light being presented herein (e.g., with supporting expert testimony and in combination with *Wookey*).

Second, as the Board previously found, *Eggert*'s protocol is fundamentally different from that discussed in RFC 5482 because, in contrast with RFC 5482, *Eggert*'s protocol calls for a common negotiated value for the timeout. Google -845 IPR, Paper 16 at 14–17. While *Eggert* discloses a negotiation protocol, RFC 5482 was purposefully designed such that “an exchange of UTO [user timeout] options between both ends of a connection is not a binding negotiation.” (EX1016, 4.)

Eggert also has meaningful disclosures that do not appear in RFC 5482 such as section 2.1 (including Figure 2). For example, it is *Eggert*'s mechanisms disclosed with reference to Figure 2 regarding exchange of ATO information during the three-way handshake that are relied upon to teach the claimed identify, generate,

Petition for *Inter Partes* Review
Patent No. 10,075,564

send, and for use steps. (*Supra* Sections IX.A.1.c–f.) In contrast, RFC 5482 “does not define a mechanism to negotiate support of the TCP User Timeout Option [UTO] during the three-way handshake.” (EX1016, 9.) Therefore, these “material differences between the asserted art and the prior art involved during examination” weigh against discretionary denial. *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8, 17-18 (Dec. 15, 2017) (precedential as to §III.C.5, first paragraph); *see also* Google -845 IPR, Paper 16 at 16–17. Thus, *Eggert* is not cumulative to RFC 5482.

Accordingly, Petitioner requests that the Board institute review.

B. The Related Litigation Provides No Basis For Discretionary Denial

The Board should not exercise its discretion to deny institution under 35 U.S.C. § 314(a). *NHK Spring Co., Ltd. v. Intrix-Plex Techs, Inc.*, IPR2018-00752, Paper 8 (Sept. 12, 2018) does not apply here as no trial date has been scheduled in the related litigation. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 3 (Mar. 20, 2020) (precedential) (“*NHK* applies ... where the district court has set a trial date to occur earlier than the Board’s deadline to issue a final written decision in an instituted proceeding.”). The six-factor test addressed in *Fintiv* (“*Fintiv* factor(s)”) favors institution. *See id.*, 5–6.

Petition for *Inter Partes* Review
Patent No. 10,075,564

The **first *Fintiv* factor** (stay) is neutral, if not weighing in favor of institution. While there is currently no stay pending IPR resolution before the Western District of Texas (“WDTX”) court, Google intends to pursue such a stay should institution be granted. The Board should not speculate as to how the court would rule on the requested stay based on actions taken in different cases. *Western Digital Corp. et al. v. Martin Kuster*, IPR2020-01391, Paper 10 at 8-9 (February 16, 2020) (finding this factor neutral despite Patent Owner assertions regarding WDTX Judge Albright’s past practices regarding stays). Moreover, Google filed a motion to transfer in the district court litigation. The Court has currently stayed the litigation pending resolution of Google’s transfer motion. (EX1035.)

The **second *Fintiv* factor** (proximity of trial dates) weighs strongly in favor of institution. *First*, the district court has not set a trial date, which “weighs significantly against exercising [] discretion to deny institution of the proceeding.” *Google LLC v. Uniloc 2017 LLC*, IPR2020-00441, Paper 13 at 35 (July 17, 2020); *Nested Bean, Inc. v. Big Beings USA PTY LTD*, IPR2020-01234, Paper 15 at 14 (Jan. 25, 2021) (same). In fact, while the parties submitted proposed schedules, the court granted Google’s motion to stay the case pending resolution of Google’s transfer

Petition for *Inter Partes* Review
Patent No. 10,075,564

motion and declined to set a case schedule.²⁸ (EX1035; EX1029.) Accordingly, determining at this time whether the Board’s anticipated FWD date (e.g., Sept.-Oct. 2022) will occur after trial in the related litigation would be speculative.

Even if the parties’ most recent proposed schedule was considered as a baseline for comparison (which Google submits would be improper at this time), dates in that proposed schedule would need to be delayed because they were proposed prior to the court’s recent stay order and did not take the stay into consideration. (EX1035; EX1029, 5-9.) Accordingly, the proposed dates in light of the court’s stay order demonstrates that trial will likely occur after August 2022.²⁹ (EX1029, 6, 9.) Thus, any projected trial date (*albeit* speculative) would be more

²⁸ Disposition of Google’s transfer motion has taken priority over other activities. (EX1035.) *See also In re Apple Inc.*, 979 F.3d 1332, 1337 (Fed. Cir. 2020) (“disposing of a filed transfer motion ‘should unquestionably take top priority.’”).

²⁹ Consistent with the court’s practice, the proposed order stages deadlines and trial date based on the proposed August 2021 *Markman* hearing and that the “actual trial date” may materially differ from the schedule and that the court would consider “reasonable amendments to the case schedule post-*Markman*.” (EX1029, 9 n.3; *id.*, 6.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

likely after the expected due date of the Board’s FWD (e.g., around September-October 2022).

The Board has instituted trial even where an *actual* litigation trial date precedes the FWD deadline by two months or more. *Sand Revolution II, LLC v. Cont’l Intermodal Grp–Trucking LLC*, IPR2019-01393, Paper 24 at 8-10 (June 16, 2020) (“*Sand*”) (informative) (factor 2 favored institution despite WDTX trial preceding FWD deadline by five months); *Western Digital Corp.*, IPR2020-01391 at 9-10 (factor 2 found to be neutral despite WDTX trial predating FWD by three and a half months); *Apple Inc. v. Parus Holdings, Inc.*, IPR2020-00686, Paper 9 at 11–13, 22 (instituting trial where WDTX trial was two months before FWD deadline); *SMIC, Am. v. Innovation Foundry Techs., LLC*, IPR2020-00786, Paper 10, at 20–21 (Oct. 5, 2020) (same, where WDTX trial date preceded FWD by one month); *Fintiv*, IPR2020-00019, Paper 11 at 5, 9 (“an early trial date” is “non-dispositive” and simply means that “the decision whether to institute will likely implicate other factors”).

It is also unlikely that trial in WDTX will proceed without delay. WDTX has issued a suspension order every month for the past eleven months suspending almost all trials in the district from March 13, 2020 to at least March 31, 2021 due to the COVID-19 pandemic, creating a large backlog of trials. (*See* EX1030; *see also*

Petition for *Inter Partes* Review
Patent No. 10,075,564

EX1031, 2 (agreeing to stipulation to postpone trial date “to allow the COVID-19 situation to ameliorate”); EX1034 (showing J. Albright’s docket including 792 cases filed in 2020).) Thus, as the Board has recognized, civil trials in WDTX “may possibly slip ... [due to] months of backlogged trials, including many active criminal cases that would take precedence over civil trials.” *HP Inc. v. Slingshot Printing LLC*, IPR2020-01085, Paper 12 at 7 (Jan. 14, 2021); *Sand* at 8-10.

Finally, the potential for transfer based on Google’s transfer motion further reduces the likelihood that trial will precede the Board’s FWD. *Dish Network, L.L.C. v. Broadband iTV, Inc.*, IPR2020-01267, Paper 15 at 17-18 (Jan. 21, 2021) (explaining that the second *Fintiv* factor “is a proxy for the *likelihood* that the trial court will reach a decision on validity issues before the Board reaches a [FWD]”). Indeed, transfer of the WDTX case to the proper venue (the Northern District of California) will undoubtedly delay any trial beyond the anticipated FWD due date. *Cf. Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 8, at 7 n.7 (Mar. 27, 2020) (denying institution based in part on scheduled trial date that was later vacated upon transfer); *see Uniloc 2017 LLC v. Google LLC*, Case No. 18-CV-00502, Dkt No. 277 (E.D. Tex. June 19, 2020) (transfer order vacating unreached deadlines).

The **third *Fintiv* factor** (investment in parallel proceedings) also weighs in favor of institution. The case is in earliest stages, and as discussed, all deadlines

Petition for *Inter Partes* Review
Patent No. 10,075,564

other than those concerning Google's transfer motion have been stayed. Thus, investment by the parties in invalidity and claim constructions issues will be delayed, especially if the case is transferred. And even if the transfer motion is denied, using the above-discussed proposed schedule as an exemplary baseline (which will require adjustments), any *Markman* hearing in the case is likely not to occur until August 2021 or later, and thus would occur near the expected due date of the Board's institution decision. (EX1029, 6.) Discovery will not commence, and dispositive motions will not be due, until after the *Markman* hearing. Other case activity that has taken place or will eventually take place (outside claim construction) that does not relate to validity should "not weigh in [the Board's] consideration of this issue." *Western Digital Corp.*, IPR2020-01391 at 11. Such minimal investment is insufficient to support discretionary denial. *Juniper Networks, Inc. v. Huawei Digital Techs. (Cheng Du) Co., Ltd.*, IPR2020-01130, Paper 13 at 12-13 (Jan. 22, 2021) (factor 3 weighs against discretionary denial when claim construction order has been entered, but fact and expert discovery has not closed, and dispositive motions and briefings have not been submitted); *Dish* at 19-21 (similar); *HP* at 7 (similar). Additionally, Google's diligence in filing this Petition just three months

Petition for *Inter Partes* Review
Patent No. 10,075,564

after receiving PO's narrowed list of asserted claims³⁰ further weighs against discretionary denial. *Dish* at 20-21 (petitioner's diligence in filing the petition weighed against discretionary denial); *Fintiv* at 11.

The **fourth *Fintiv* factor** (overlap) is neutral if not in favor of institution. There is, at this early stage of the district court litigation, no evidence of overlap with the grounds here and any invalidity positions to be pursued in the district court. Indeed, preliminary invalidity contentions have not yet been served. And, as discussed above, the district court must first resolve the pending transfer motion before addressing substantive issues, including invalidity. Given the stay of the litigation, it is likely expert discovery relating to invalidity will not be completed until at the earliest mid-2022 and dispositive motions on those issues would be due

³⁰ PO initially asserted 455 claims across eight patents (including 29 claims from the '564 patent). (EX1017, 1.) The WDTX court ordered PO to substantially reduce the asserted claims to approximately 65 claims. On October 20, 2020, PO narrowed its asserted claims as ordered, but "reserve[d] the right to either narrow the claims further, **substitute claims**, or **add a reasonable number of claims**." (EX1032.) On December 4, 2020, PO informed Petitioner that it will not modify its narrowed list of asserted claims "**at this time**." (EX1033.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

after that. Accordingly, substantive invalidity issues would not be addressed in the litigation until well past the institution decision and likely very near the time of the Board's expected FWD. Moreover, Petitioner may rely on invalidity grounds not asserted here in the district court litigation, including system art and prior art references and combinations not raised herein. Therefore, any potential overlap between this proceeding and the district court is minimal and speculative at this time.

Other circumstances (*Fintiv* factor six) also favor institution. The Petition presents strong grounds demonstrating the unpatentability of the challenged claims, which weighs against discretionary denial. (*See supra* Section IX.) *Western Digital Corp.*, IPR2020-01391 at 14-15. Indeed, the Board has already instituted the Google -845 IPR and the Unified -742 IPR challenging claims of the related '995 patent. (*See supra* Section II.) The Google -845 IPR is based in part on *Eggert*, which is being applied here. Moreover, this petition and the concurrently-filed sister petition (*id.*) are the only challenges to the '564 patent before the Board, which is a "crucial fact" favoring institution. *Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 10 at 6 (May 12, 2020).³¹

³¹ Discretionary denial under *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper No. 19 is not applicable here given the

Petition for *Inter Partes* Review
Patent No. 10,075,564

While ***Fintiv* factor 5** (same parties) may weigh slightly in favor of denial, the remaining factors are at least neutral, if not strongly favoring institution. Further, even if the Board finds that **factor 1** (or even **factor 2**) somehow tilts in favor of denial, the investment of resources, lack of evidence of overlap, and/or strength of Petitioner’s unpatentability positions (**factors 3, 4, 6**) outweigh these other factors. Accordingly, based on a “holistic view” of whether integrity of the system and efficiency is best served, institution here is proper. *Samsung Elecs. Co. Ltd. v. Dynamics Inc.*, IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

’564 patent is not at issue in any other proceeding pending before the Board (outside of Petitioner’s concurrently-filed sister petition, *supra* Section II).

Petition for *Inter Partes* Review
Patent No. 10,075,564

XII. CONCLUSION

For the reasons given above, Petitioner requests institution of IPR for the challenged claims based on each of the specified grounds.

Respectfully submitted,

Dated: March 15, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,075,564

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,075,564 contains, as measured by the word-processing system used to prepare this paper, 13,966 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: March 15, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,075,564

CERTIFICATE OF SERVICE

I hereby certify that on March 15, 2021, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,075,564 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on PAIR:

Patrick Caldwell (Reg. No. 44580)
G. Gordon (Reg. No. 64517)
P.O. Box 59655
Dept. SVIPGP
Dallas, TX 75229

A courtesy copy was also sent via electronic mail to the Patent Owner's litigation counsel at the following addresses:

TD-PTAB@devlinlawfirm.com
ddahlgren@devlinlawfirm.com
dlflitparas@devlinlawfirm.com
1526 Gilpin Ave.
Wilmington, DE 19806

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)

EXHIBIT 4

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

JENAM TECH, LLC,
Patent Owner

Patent No. 10,075,564

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,075,564**

Petition for *Inter Partes* Review
Patent No. 10,075,564

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	1
III.	PAYMENT OF FEES	2
IV.	GROUND FOR STANDING.....	3
V.	PRECISE RELIEF REQUESTED	3
	A. Claims for Which Review Is Requested	3
	B. Statutory Ground of Challenge	3
VI.	LEVEL OF ORDINARY SKILL	4
VII.	OVERVIEW OF THE '564 PATENT	5
VIII.	CLAIM CONSTRUCTION	8
IX.	DETAILED EXPLANATION OF GROUND.....	11
	A. Ground 1: The Combination of <i>Wookey</i> and <i>Berg</i> Render Obvious Claims 24–29.....	11
	1. Claim 24.....	11
	2. Claim 25.....	56
	3. Claim 26.....	59
	4. Claim 27.....	61
	5. Claim 28.....	64
	6. Claim 29.....	76
X.	DISCRETIONARY DENIAL IS NOT APPROPRIATE	78
XI.	CONCLUSION.....	86

Petition for *Inter Partes* Review
Patent No. 10,075,564

LIST OF EXHIBITS

EX1001	U.S. Patent No. 10,075,564
EX1002	Declaration of Bill Lin, Ph.D.
EX1003	Curriculum Vitae of Bill Lin, Ph.D.
EX1004	Prosecution History of U.S. Patent No. 10,075,564
EX1005	U.S. Pre-Grant Publication No. 2007/0171921 to Wookey <i>et al.</i>
EX1006	RESERVED
EX1007	U.S. Patent 6,674,713 to Berg et al.
EX1008	DARPA RFC 793 TRANSMISSION CONTROL PROTOCOL
EX1009	U.S. Patent 6,584,546 to Kavipurapu
EX1010	U.S. Patent No. 9,923,995
EX1011	Bova et al., RELIABLE UDP PROTOCOL <draft-ietf-sigtran-reliable-udp-00.txt> 25 February 1999
EX1012	U.S. Pre-Grant Publication No. 2004/0093376 to De Boor et al.
EX1013	U.S. Patent 7,535,913 to Minami et al.
EX1014	U.S. Pre-Grant Publication No. 2005/0204013 to Raghunath <i>et al.</i>
EX1015	U.S. Pre-Grant Publication No. 2007/0005804 to Rideout
EX1016	U.S. Pre-Grant Publication No. 2004/0098748 to Bo et al.
EX1017	<i>Jenam Tech, LLC's First set of Infringement Contentions</i> regarding U.S. Patent No. 10,075,564 (August 21, 2020)
EX10181	IETF RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1

Petition for *Inter Partes* Review
Patent No. 10,075,564

EX1019	U.S. Patent No. 7,636,805 to Rosenberg
EX1020	U.S. Patent No. 6,212,175 to Harsch
EX1021	U.S. Patent No. 8,259,716 to Diab
EX1022	U.S. Patent No. 6,665,727 to Hayden
EX1023	U.S. Patent No. 6,981,048 to Abdolbaghian <i>et al.</i>
EX1024	<i>Jenam Tech., LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Joint Submission of Proposed Amended Scheduling Order, ECF No. 57 (W.D. Tex. Mar. 4, 2021)
EX1025	Thirteenth Supplemental Order Regarding Court Operations Under the Exigent Circumstances Created By the COVID-19 Pandemic, The United States District Court for the Western District of Texas (February 2, 2021)
EX1026	<i>Digital Retail Apps, Inc. v. H-E-B, LP</i> , Case No. 6:19-cv-00167, Joint Stipulation and Order Postponing Trial, ECF No. 182 (W.D. Tex. Jan. 13, 2021)
EX1027	<i>Jenam Tech, LLC's</i> Preliminary Narrowing of Claims (October 20, 2020)
EX1028	<i>Jenam Tech, LLC's</i> Correspondence Regarding Narrowing of Claims (December 4, 2020)
EX1029	Judge Alan D. Albright's Case Statistics By Year (Retrieved from DocketNavigator on March 9, 2021)
EX1030	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Order Granting Motion to Stay Case, ECF No. 58 (W.D. Tex. Mar. 10, 2021)

Petition for *Inter Partes* Review
Patent No. 10,075,564

I. INTRODUCTION

Google LLC (“Petitioner”) requests *inter partes* review (“IPR”) of claims 24–29 (“the challenged claims”) of U.S. Patent No. 10,075,564 (“the ’564 patent”) (EX1001) assigned to Jenam Tech, LLC (“Patent Owner” or “PO”). For the reasons below, the challenged claims should be found unpatentable and canceled.

II. MANDATORY NOTICES

Real Parties-in-Interest: Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies the following as the real parties-in-interest: Google LLC.¹

Related Matters: The ’564 patent is asserted in the following civil actions: *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.); and *Jenam Tech, LLC v. Samsung Electronics Co., Ltd.*, Case No. 4:20-cv-00279 (E.D. Tex.) (voluntarily dismissed).

The ’564 patent claims priority to U.S. Patent Application No. 15/694,802, filed on September 3, 2017, and issued as U.S. Patent No. 9,923,995 (“the ’995 patent”). (EX1001, Cover.) The ’995 patent is subject to the following instituted

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

Petition for *Inter Partes* Review
Patent No. 10,075,564

IPRs: *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845 (“Google -845 IPR”); and *Unified Patents, LLC v. Jenam Tech, LLC*, IPR2020-00742 (“Unified -742 IPR”).

Petitioner is concurrently filing another IPR petition challenging the ’564 patent.² Petitioner is also concurrently filing IPR petitions challenging U.S. Patent Nos. 10,075,565 and 10,375,215 which are also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.).

Counsel and Service Information: Lead counsel: Naveen Modi (Reg. No. 46,224). Backup counsel: (1) Joseph E. Palys (Reg. No. 46,508), (2) Quadeer A. Ahmed (Reg. No. 60,835), (3) Jason Heidemann (Reg. No. 77,880).

Service information is Paul Hastings LLP, 2050 M Street NW, Washington, DC 20036, Tel.: 202.551.1700, Fax: 202.551.1705, email: PH-Google-Jenam-IPR@paulhastings.com. Petitioner consents to electronic service.

III. PAYMENT OF FEES

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-2613.

² Petitioner concurrently submits herewith its Notice Regarding Multiple Petitions.

Petition for *Inter Partes* Review
Patent No. 10,075,564

IV. GROUNDS FOR STANDING

Petitioner certifies that the '564 patent is available for review and Petitioner is not barred or estopped from requesting review on the ground identified herein.

V. PRECISE RELIEF REQUESTED

A. Claims for Which Review Is Requested

Petitioner respectfully requests review of claims 24–29 (“the challenged claims”) of the '564 patent and cancellation of those claims as unpatentable.

B. Statutory Ground of Challenge

The challenged claims should be canceled as unpatentable in view of the following ground:

Ground 1: Claims 24–29 are rendered obvious under pre-AIA 35 U.S.C. § 103(a) over U.S. Publication No. 2007/0171921 to *Wookey et al.* (EX1005, “*Wookey*”) in view of U.S. Patent No. 6,674,713 to *Berg et al.* (EX1007, “*Berg*”).³

The '564 patent issued from Application No. 15/915,047, filed on March 7, 2018, which is a continuation of Application No. 15/694,802, filed on September 3, 2017, which is a continuation-in-part of Application No. 14/667,642,

³ Other references discussed herein are provided to show the state of the art at the time of the alleged invention of the '564 patent.

Petition for *Inter Partes* Review
Patent No. 10,075,564

filed on March 24, 2015, which is a continuation-in-part of Application No. 13/477,402, filed on May 22, 2012, which is a continuation of Application No. 12/714, 454 (“the ’454 application”), filed on February 27, 2010. (EX1001, 1:8–28.)

For purposes of this proceeding only, Petitioner assumes, without conceding, the earliest effective filing date of the ’564 patent is February 27, 2010, (filing date of the ’454 application).

Wookey published on July 26, 2007, from an application filed on November 14, 2006. (EX1005, Cover.) *Berg* issued on January 6, 2004, from an application filed on February 23, 1999. (EX1007, Cover.) Thus, *Wookey* and *Berg* qualify as prior art at least under 35 U.S.C. § 102(b) (pre-AIA).

Wookey and *Berg* were not considered during prosecution of the ’564 patent. (EX1001 at Cover; *generally* EX1004.)

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art at the time of the alleged invention of the ’564 patent (“POSITA”) would have had an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at

Petition for *Inter Partes* Review
Patent No. 10,075,564

least two years of work experience in the field of networking. (EX1002 ¶¶16–18)⁴

More education can supplement practical experience and vice versa. (*Id.*)

VII. OVERVIEW OF THE '564 PATENT

The '564 patent is directed to networking, and in particular, the sharing of information for detecting an idle TCP connection. (EX1001, 2:26–28, 7:29–50.)

Figure 7 illustrates such a process.

⁴ Petitioner submits the declaration of Dr. Bill Lin (EX1002), an expert in the field of the '564 patent. (EX1002 ¶¶3–15; EX1003.)

Petition for *Inter Partes* Review
 Patent No. 10,075,564

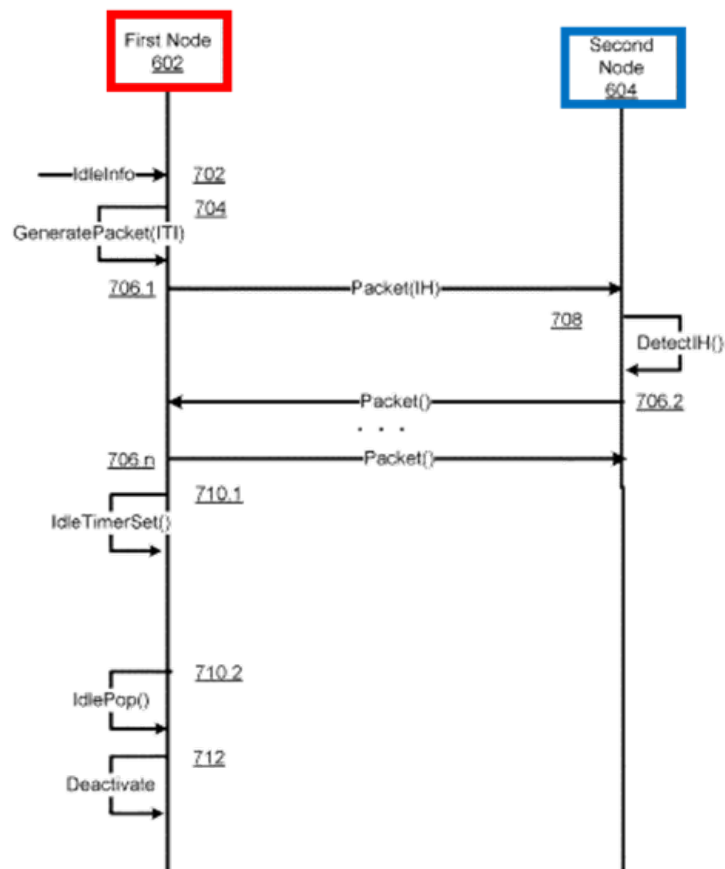


Fig. 7

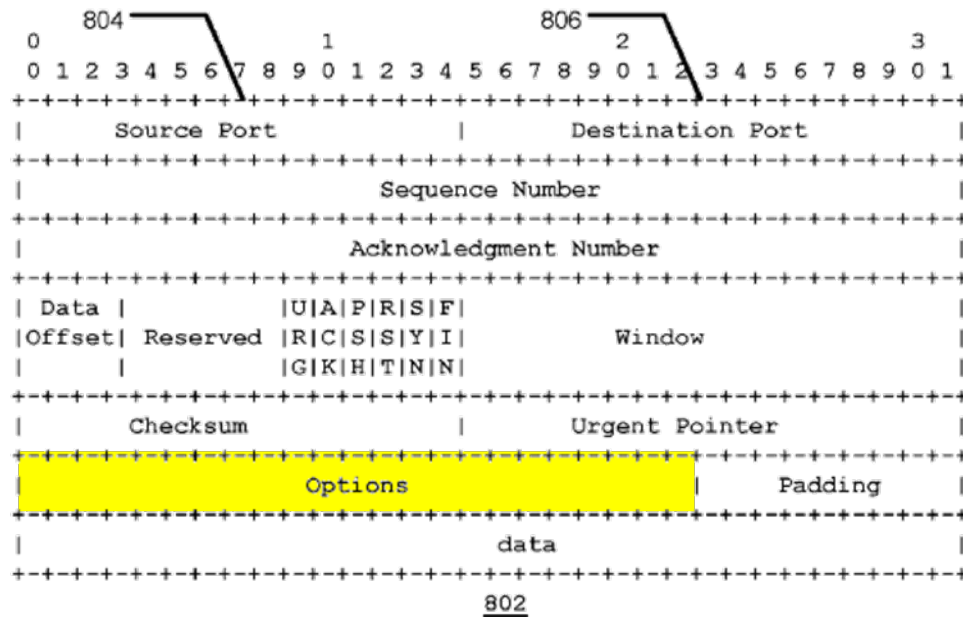
(*Id.*, FIG. 7 (annotated); EX1002 ¶64.)

First, the **first node 602** receives a message 702 that identifies idle information representing a duration for an idle time period (“ITP”). (EX1001, 10:55–57.) Message 702 can be “a message received via a network.” (*Id.*, 10:57–63.) The idle information “may include and/or identify a duration of time for

Petition for *Inter Partes* Review
Patent No. 10,075,564

detecting an idle time period.” (*Id.*, 11:37–41.) The duration “may be specified according to various measures of time.” (*Id.*; EX1002 ¶65.)

Next, the **first node** “generat[es] a TCP packet including an ITP header based on received idle information.” (EX1001, 14:46–49.) The **TCP options field** of a TCP packet may store this ITP header (IH). (*Id.*, 14:24–29.) Figure 8, “adapted from RFC 793,” illustrates this configuration. (*Id.*, FIG. 8, 5:28–30, 14:46–49.)

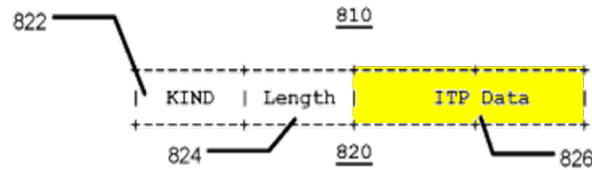


(*Id.*, FIG. 8 (cropped and annotated); EX1002 ¶66.)

The **TCP options field** contains a KIND sub-field that identifies the type of option presented, a length sub-field that specifies the length of the option field, and an **ITP Data sub-field** containing data. (EX1001, FIG. 8.) The **TCP options field** carries the IH. (*Id.*, 14:46–49.) The IH can also be “in structures and locations [in

Petition for *Inter Partes* Review
Patent No. 10,075,564

a TCP packet] other than those specified for TCP options in RFC 793” (*id.*, 14:55–58).



Figures are adapted from RFC 793

Fig. 8

(*Id.*, FIG. 8 (cropped and annotated); EX1002 ¶¶66.)

The IH is exchanged during the three-way handshake (EX1001, 13:49–52.) For example, the **first node** transmits a message 706.1, i.e., a TCP packet including an IH to the **second node 604**. (*Id.*, 15:29–31.) Message 708 exemplifies the **second node**’s detection of the IH in the received TCP packet. (*Id.*, 20:50–53; EX1002 ¶¶67–72; *see also id.* ¶¶19–63 (discussing technology background), citing, e.g., EXS. 1008–1009, 1011–1016, 1018–1023.)

VIII. CLAIM CONSTRUCTION

Under the applicable standard in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), claim terms are typically given their ordinary and customary meanings as understood by a POSITA at the time of the invention based on the claim language, specification, and the prosecution history of record. *Phillips*, 415 F.3d at

Petition for *Inter Partes* Review
Patent No. 10,075,564

1313; *see also id.* at 1312–16. The Board, however, only construes the claims when necessary to resolve the controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper No. 11 at 16 (Aug. 14, 2015) (citing *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)). Petitioner believes that no express constructions of any claim terms are necessary to assess whether the prior art reads on the challenged claims.⁵ (EX1002 ¶73.)

Claims 24 and 28 recite the term “non-TCP packet” (EX1001, 25:27–53, 28:6–29) and claims 24–29 recites the term “non-TCP connection” (*e.g., id.*, 26:27–27:37), which are also recited in claims of the related ’995 patent at-issue in the Unified -742 IPR (*see supra* Section II). The ’564 patent and the ’995 patent share a common specification. (*Compare* EX1001 *with* EX1010 (’995 patent).)

⁵ Petitioner reserves all rights to raise claim construction and other arguments, including challenges under 35 U.S.C. §§ 101 or 112, in district court as relevant to those proceedings. *See, e.g., Target Corp. v. Proxicom Wireless, LLC*, IPR2020-00904, Paper 11 at 11–13 (November 10, 2020). A comparison of the claims to any accused products in litigation may raise controversies not presented here given the similarities between the references and the patent.

Petition for *Inter Partes* Review
Patent No. 10,075,564

The '564 patent discloses the term “non-TCP” only in the “Summary” section (EX1001, 3:7–52), which explains is “not an extensive overview of the disclosure and it does not identify key/critical elements ... or delineate the scope of the invention” (EX1001, 2:34–36). As discussed below, *Berg*’s RUDP protocol is a non-TCP protocol and the RUDP packets exchanged are non-TCP packets. (*See, e.g., infra* Section IX.A.1.d.) Moreover, the Board preliminarily found that *Berg* discloses these non-TCP limitations. *See, e.g.,* Unified -742 IPR, Paper 11 at 10–11.

Thus, because Petitioner relies on *Berg*’s similar disclosures to satisfy the “non-TCP” terms herein (*see infra* Section IX.A), and given that the '564 patent offers no evidence requiring a special meaning of these terms and that the prior art discloses these features under any reasonable interpretation, construction of these terms is unnecessary.

IX. DETAILED EXPLANATION OF GROUND**A. Ground 1: The Combination of *Wookey* and *Berg* Render Obvious Claims 24–29****1. Claim 24****a) A non-transitory computer readable medium, comprising:**

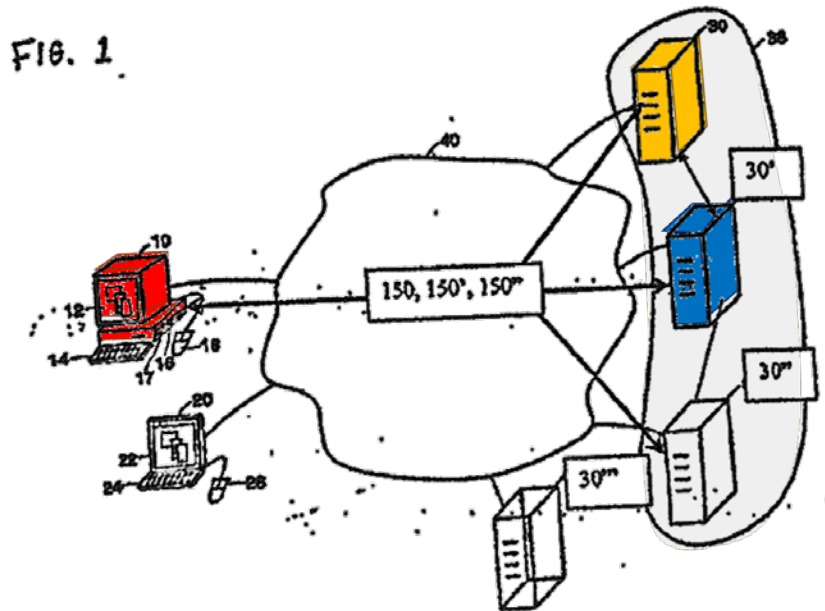
To the extent limiting, *Wookey* discloses the limitations of this preamble. (EX1002 ¶¶98–111.) For example, *Wookey* discloses a **remote machine 30** that includes a main memory unit 104 and cache memory 140 (individually or collectively, the claimed “non-transitory computer readable medium”). (*Id.*)

Figures 1, 2C, and 3D below show an exemplary environment in which a **client machine 10**, **remote machine 30**, and **remote machine 30'** are employed to permit a **client machine 10** access to a resource.⁶ (*E.g.*, EX1005 ¶¶[0016], [0020],

⁶ *Wookey*'s disclosures (including its figures) include certain typographical errors when referencing its client machines (referring to them interchangeably as 10, 10' or 20). (EX1002 ¶105; EX1005, ¶¶[0135]–[0136], [0152]–[0154], [0192]–[0196], [0207]–[0216], FIGS. 1, 2C, 3D.) For example, the bottom figure in Fig. 2C erroneously shows client machine “20,” but the corresponding disclosure for both figures in Fig. 2C describes client machine “10.” (EX1005 ¶¶[0192]–

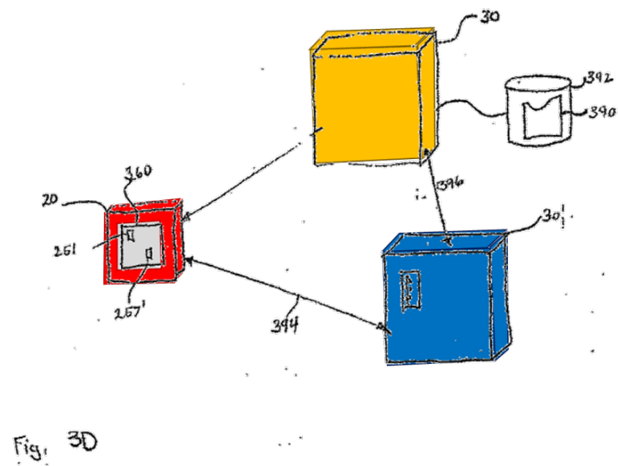
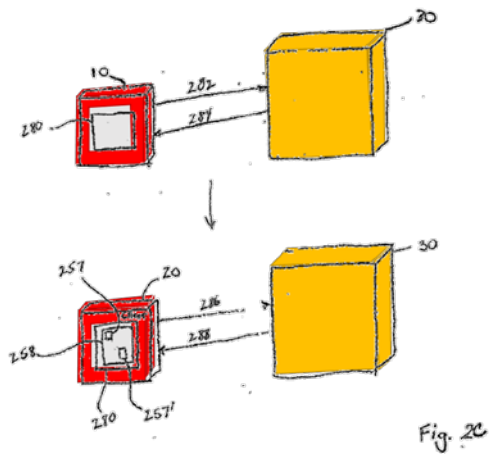
Petition for *Inter Partes* Review
 Patent No. 10,075,564

[0196]; EX1002 ¶¶74–83 (*Wookey* overview).) These client and remote machines are “typical computers.” (*Id.* ¶¶[0021], FIGS. 1A–1B.)



[0196].) Moreover, in context of *Wookey*'s disclosures, a POSITA would have understood that any teachings regarding one client machine 10/10'/20 in *Wookey* apply equally to the other client machines. (EX1002 ¶105.)

Petition for *Inter Partes* Review
Patent No. 10,075,564



(*Id.*, FIGS. 1, 2C, 3D (each annotated); EX1002 ¶¶101–104.)

Figures 1A and 1B below depict typical computer architectures for both client and remote machines, where each computer 100 includes a CPU 102, main memory unit 104, and cache memory 140. (EX1005 ¶[0021]; *see also id.*, ¶¶[0161]–[0165], FIG. 33.)

FIG. 1A

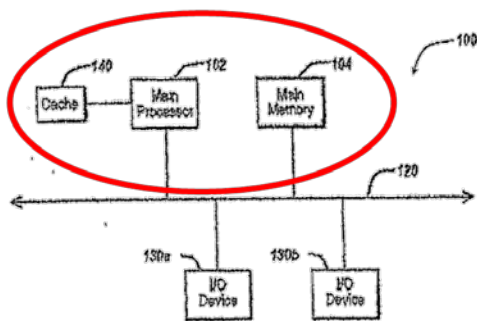
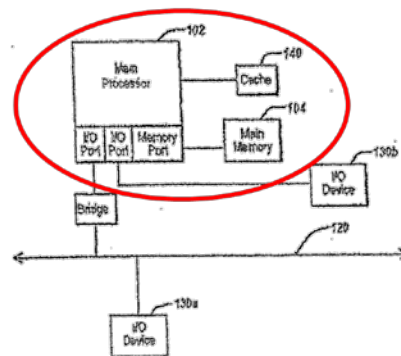


FIG. 1B



(*Id.*, FIGS. 1A and 1B (annotated); EX1002 ¶106.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

The main memory unit 104 includes memory chips (e.g., SRAM) that store data/instructions accessed by CPU 102. (EX1005 ¶¶[0162]–[0163].) As was known in the art at the time, cache memory often stores data/instructions accessible to a CPU. (EX1002 ¶¶107–109; EX1009, 1:15–39.) Cache memory 140 and main memory unit 104 of **remote machine 30** are each a non-transitory memory device and thus individually or collectively disclose the claimed non-transitory computer readable medium. (EX1002 ¶¶109–111; *infra* Sections IX.A.1(b)–(f).)

b) Claim 24.b

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation, which is addressed below in two parts. (EX1002 ¶¶112–114.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

- (1) **code for use by a client node [], where the code, when used by the client node, results in the client node utilizing the network application to operate in accordance with the non-TCP protocol to:**⁷

As explained below, *Wookey* discloses that **remote machine 30** causes encoded Uniform Resource Locators (URLs) (“code”) associated with icons on an HTML page 288 (where the HTML page 288 is also “code”)⁸, which is stored in its non-transitory memory, to be sent to a **client machine 10** (“client node”) for use by **client machine 10**. (EX1005 ¶¶[0192]–[0193]; *supra* Section IX.A.1.a.)⁹ Also explained below, the *Wookey-Berg* combination discloses and/or suggests that the code, when used by **client machine 10**, causes **client machine 10** to communicate with a different **remote machine 30**’ using a non-TCP protocol. (*See, e.g.*, EX1005 ¶¶[0196], [0207], [0213], [0216]; EX1002 ¶¶115–116.)

Wookey discloses, regarding Figure 2C, that **client machine 10** executes a web browser application 280 (e.g., a “network application”) (EX1005 ¶[0192]) that

⁷ The ’564 patent specification does not provide any disclosure describing the claimed “code” features. (EX1002 ¶112 n.6.) Therefore, while Petitioner addresses this limitation with respect to the asserted prior art, Petitioner does not concede that the ’564 patent provides adequate disclosures of such claimed features.

Petition for *Inter Partes* Review
Patent No. 10,075,564

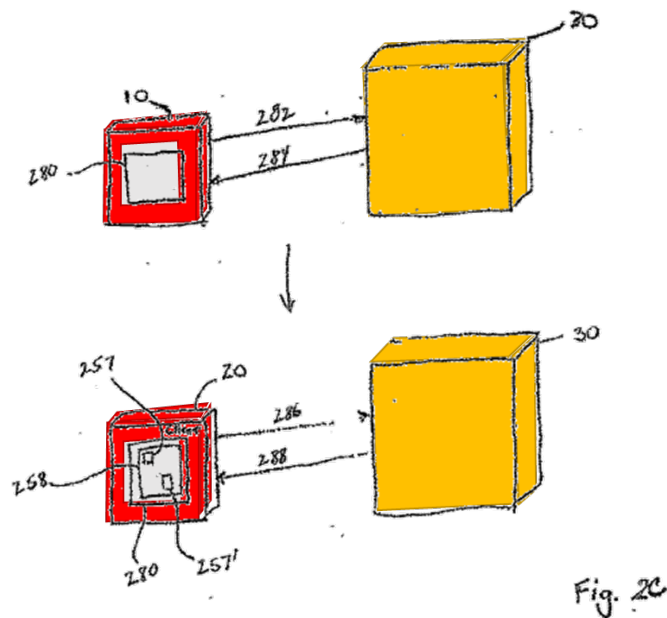
“transmits a request 282 to access URL address corresponding to an HTML page residing on” **remote machine 30** (*id.* ¶[0193].) (*See also id.*, ¶¶[0020], [0024], [0213].)¹⁰

⁸ Petitioner’s analysis regarding the “code” limitation equally applies whether the “code” is the encoded URLs in the HTML page or the HTML page itself (which includes the encoded URLs). (EX1002 ¶113 n.7.)

⁹ *Wookey* uses the term “machine” to describe what the ’564 patent refers to as a “node.” (EX1005 ¶¶[0153], [0301]; EX1002 ¶113 n.8.)

¹⁰ **Client machine 10** may use a TCP protocol for establishing a TCP connection with **remote machine 30**. (EX1005 ¶¶[0732], [0156]; *see infra* Section IX.A.1.f.)

Petition for *Inter Partes* Review
 Patent No. 10,075,564



(*Id.*, FIG. 2C (annotated); EX1002 ¶116.)

Remote machine 30, in turn, transmits an HTML page 288, that contains encoded URLs, to **client machine 10** (“client node”), where the HTML page 288 is stored in the memory (e.g., main memory unit 104 and/or cache memory 140) of **remote machine 30**. (EX1005 ¶[0193]; *see also id.* ¶¶[0207], [0213], [0754], [0757], FIG. 27; EX1002 ¶¶117–118.)

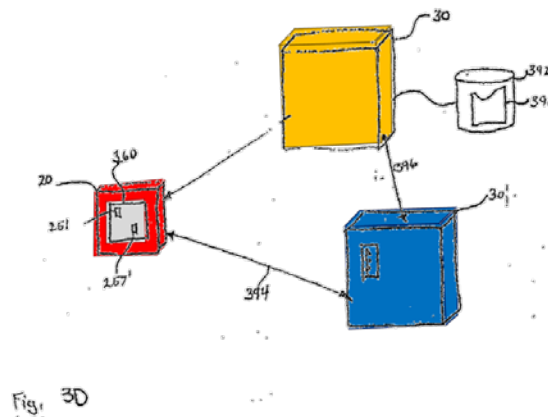
The HTML page 288 includes encoded URLs associated with icons (e.g., 257, 257’), and at least these encoded URLs are the claimed “code for use by a client node.” (EX1002 ¶¶117–119.) Indeed, “[e]ach icon 257, 257’ is associated with *an encoded URL that specifies: the location of the resource ...; a launch command*

Petition for *Inter Partes* Review
Patent No. 10,075,564

associated with the resource; and a template identifying how the results of accessing the resource should be displayed.” (EX1005 ¶[0216].)¹¹ Each encoded URL “*contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (*Id.*) Thus, when a user *clicks* an icon 257, 257’ from the displayed HTML page 288, **client machine 10** utilizes the web browser 280 to use the encoded URL associated with the clicked icon to initiate the process for establishing a second connection (e.g., connection 394 in Figure 3D below) with **remote machine 30’**. (*Id.*) Accordingly, the encoded URLs are the claimed “code for use by a client node” because the URLs are instructions that, when used by **client machine 10**, command **client machine 10** to access resources from one or more remote machines.

¹¹ Emphasis added unless otherwise noted.

Petition for *Inter Partes* Review
 Patent No. 10,075,564



(*Id.*, FIG. 3D (annotated); EX1002 ¶¶119–122.)

The above understanding is consistent with the '564 patent description of “code.” (EX1001, cl. 30 (27:6) (indicating code “includes one or more links”).) Indeed, *Wookey*’s encoded URLs associated with the icons on the HTML page 288 would similarly include a link (e.g., specifying the remote resource location and launch command to connect to the resource). (EX1002 ¶123.)

Moreover, HTML page 288 (including the encoded URLs and icons) also discloses the claimed “code for use by a client node” because an HTML page contains HTML code used to define, e.g., what is displayed on the screen of **client machine 10**. (EX1002 ¶124; EX1012 ¶¶[0022], [0024], [0026], [0079].) The display would have included the icons associated with the URLs that **client machine 10** uses to access the desired resource. This understanding concerning an HTML page containing code also tracks PO’s interpretation of this limitation in its

Petition for *Inter Partes* Review
Patent No. 10,075,564

infringement positions for the '564 patent. (EX1017, 83 (“code (e.g., HTML pages, etc.) for use by a client node (e.g., device that receives the HTML pages, etc.)”).)¹²

Thus, for these reasons, *Wookey* discloses, “code for use by a client node [], where the code, when used by the client node, results in the client node utilizing the network application to operate [] to” establish a connection (e.g., the connection 394), as claimed. (EX1002 ¶125.)

Regarding the connection established between **client machine 10** and **remote machine 30**, *Wookey* provides that **client machine 10** includes an “HTTP client agent,” such as a web browser application 280, which “can use any type of protocol.” (EX1005 ¶[0159]; *see also id.* ¶[0155], [0192]–[0195].) For example, connection 394 can be made using various protocols, including the Virtual Network Computing (VNC) protocol. (*Id.* ¶[0216].) VNC is a desktop-sharing system that allows a computer to remotely access and control another computer. (EX1002 ¶126; EX1014 ¶[0019].) VNC can run over various transport protocols, including “TCP/IP, IPX/SPX, and NetBEUI.” (EX1005 ¶[0225]; *see also id.* ¶¶[0155], [0216].)

¹² Petitioner’s references to PO’s infringement allegations are not indicative of any concession to PO’s allegations that any accused instrumentality infringes any limitation of any claim of the challenged patent.

Petition for *Inter Partes* Review
Patent No. 10,075,564

IPX/SPX and NetBEUI protocols are examples of non-TCP protocols. (EX1002 ¶¶127–128; EX1013, 1:20–33.)

Wookey further explains that **client machine 10** may use “a different type of protocol than the one used to send the request to the remote machine 30.” (EX1005 ¶[0732]; EX1002 ¶129.)

Wookey also identifies desired characteristics associated at least with the second connection that a POSITA would have considered when choosing a protocol such as negotiating parameters related to the connection, reducing unintentional termination of sessions due to an imperfect connection, detecting and handling disconnections (like when a mobile device enters an elevator), and specifying an inactive time before connection termination. (EX1002 ¶130; EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153].)

Wookey further explains that machines **10** and **30**’ “exchange a set of messages which negotiate the parameters under which communications will occur.” (EX1005 ¶[0774].) Once negotiations are complete, **client machine 10** and **remote machine 30**’ can communicate over a network application. (*Id.*; see also *id.* ¶¶[0738]–[0739], [0744], [0772]–[0773]; EX1002 ¶131.)

Moreover, when **client machine 10** uses one of the encoded URLs (e.g., the one pointing to **remote machine 30**’) associated with one of the icons on the HTML

Petition for *Inter Partes* Review
Patent No. 10,075,564

page 288, it results in **client machine 10** utilizing the web browser application to establish a connection (e.g., connection 394) with **remote machine 30'** using a transport protocol. (EX1002 ¶132.)

While *Wookey* discloses that various protocols, including non-TCP protocols (e.g., IPX/SPX), may be used to establish the second connection (EX1005, ¶[0225]) (including its desired characteristics), *Wookey* does not expressly provide details regarding establishing and maintaining the second connection or the types of negotiable parameters used to establish the second connection. (EX1002 ¶133.) However, it would have been obvious to a POSITA to consider and implement a protocol following *Wookey*'s desired characteristics (e.g., such as those identified above) for the second connection between **client machine 10** and **remote machine 30'**, which may be “a different type of protocol than the one used to send the request to the remote machine 30.” (EX1005 ¶[0732].)

Berg describes such a protocol (a “*non-TCP protocol*”) that allows nodes to negotiate or modify timeout timers on a per-connection basis to provide a reliable and efficient network connection. Accordingly, based on the teachings of *Berg* and the knowledge of a POSITA and guidance by *Wookey*, it would have been obvious to configure *Wookey*'s **client machine 10** such that when the “code” (e.g., the encoded URLs in the HTML page 288 or the HTML page 288 including the encoded

Petition for *Inter Partes* Review
Patent No. 10,075,564

URLs) is used, **client machine 10** uses the network application (browser application 280) to operate in accordance with a protocol such as the one described by *Berg* having at least some characteristics contemplated by *Wookey* for such a second connection between **client machine 10** and **remote machine 30'**. (EX1002 ¶134.)

(a) Berg

Berg, like *Wookey*, relates to establishing a reliable connection between a client and server as it states that “[t]o provide a reliable backhaul, it is important to be able to maintain multiple IP connections or sessions between the signaling terminal device and the call processing device, so that message[s] can be transmitted even when the inherently unreliable IP network fails” and that “[t]he system must be configured so that communications are not interrupted upon network failure, and so that communications can resume when the network comes back up.” (EX1007, 2:4–11; *see also id.*, 1:16–32, 1:54–64, 1:65–2:3, 2:43–47, FIGS. 1A–B; EX1002 ¶¶84–97 (*Berg* overview).) Thus, *Berg* discloses networking features in the same technical field as *Wookey*. (EX1002 ¶¶135–136; *see also infra* Section IX.A.1.b.1.b (discussing similarities between *Wookey* and *Berg*).)

Berg provides a gateway device (acting as a client) connected to a media gateway controller (acting as a server). (EX1007, 3:14–23, 2:55–60.) In *Berg*’s system, both the client and server execute a session manager to manage the network

Petition for *Inter Partes* Review
 Patent No. 10,075,564

communications. (EX1007, 2:54–60, 7:63–8:2, 8:24–31.) The session manager operates “above a *reliable communication layer*” (*Id.*, 2:54–60), which “determines when or if a session is connected or failed” (*id.*). *Berg* explains that “[a] protocol layer is ‘reliable’ when it guarantees that it will deliver or generate an error message for every message that an application program requests the transport layer using that protocol to transport.” (*Id.*, 8:27–31; *see also id.*, 8:11–13.) *Berg*’s Figure 2 below describes the layers that computer systems use to communicate over a network, which is based on the Open Systems Interconnection (OSI) reference model. (*Id.*, 8:3–17, 5:27–55.)

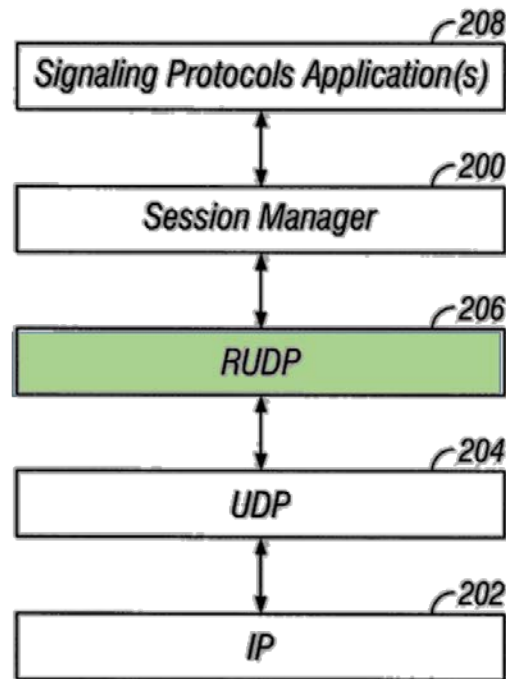


FIG. 2

(*Id.*, FIG. 2 (annotated); EX1002 ¶¶137–138.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

Berg explains that a **Reliable User Datagram Protocol (RUDP) 206** runs on top of the User Datagram Protocol (UDP) protocol software 204. (EX1007, 8:10–17.) The UDP is a non-TCP communications protocol that is primarily used for establishing low-latency and loss-tolerating connections, especially time-sensitive transmissions between applications. (EX1002 ¶¶139–140; EX1015 ¶[0110]; EX1016 ¶[0010].) The RUDP layer 206 allows *Berg*’s system to determine “when or if a session is connected or failed.” (EX1007, 2:54–60.)¹³ Above the **RUDP layer 206** in Figure 2 is the Session Manager 200 and Signaling Software Application(s) 208. (*Id.*, 8:24–27.) The Session Manager 200 can run above any reliable communication mechanism (e.g., RUDP 206). (*Id.*)

Berg notes that the “TCP/IP has a number of characteristics that make it unsuitable for” some applications. (*Id.*, 17:28–30.) In particular, “[m]ost TCP/IP implementations that allow properties like timers to be modified, *do not allow* the modification to be done on a per-connection basis.” (*Id.*, 17:40–42; EX1002 ¶141.)

In contrast to TCP/IP implementations, RUDP allows the “*characteristics of each connection to be individually configured* so that many protocols with different

¹³ *Berg* uses “RUDP layer 206” and “communication layer” to refer to the same layer in Figure 2. (EX1002 ¶139 n.13; EX1007, 2:54–60, 8:13, 8:24–28.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

transport requirements can be implemented simultaneously on the same platform.” (EX1007, 17:42–46.) For example, RUDP allows the negotiation of various parameters including timeout values associated with the retransmission timeout (*id.*, 20:24–29, 22:44–56), acknowledgement timeout (*id.*, 20:29–35, 23:15–24), null segment (*id.*, 20:36–41, 23:45–59), and transfer state (*id.*, 20:42–47, 24:15–25). (EX1002 ¶142.)

Further, the “RUDP is a lightweight protocol layer designed to run on top of UDP [that] can provide reliable in-order delivery” (EX1007, 16:66–67; *see also id.*, 17:9–16) and it “has a *very flexible design* that would make it *suitable for a variety of transport uses*. (*Id.*, 17:13–15.) In an RUDP connection, *Berg* explains that the messages sent are UDP packets (not TCP packets) that include an RUDP header. (EX1007, 17:59–18:20, 19:1–40; EX1002 ¶143; *see also infra* Section IX.A.1.c.)

Thus, the RUDP is a non-TCP protocol. (EX1002 ¶144.) *See also* Unified - 742 IPR, Paper 11 at 10 (Board agreeing that *Berg*’s RUDP is a non-TCP protocol).

(b) Reasons to Combine

Based on *Wookey*’s and *Berg*’s disclosures, the POSITA’s knowledge, and the discussions above, it would have been obvious to configure and implement *Wookey*’s **client machine 10** (including its web browser application 280) to use a protocol like *Berg*’s RUDP (“non-TCP protocol”) in order to set up an RUDP

Petition for *Inter Partes* Review
Patent No. 10,075,564

(“second protocol”) connection with **remote machine 30’** (*see, e.g.*, EX1005, Figure 3D). (EX1002 ¶145.) Such an implementation would have had many advantages and been a predictable and straightforward result achieved by combining well-known technologies using known methods. *See KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 416–18 (2007).

As explained, both *Wookey* and *Berg* are directed to establishing a reliable connection between two nodes, where one or both nodes may experience an unintentional disconnection, and thus disclose features in a similar technological field. (EX1005 ¶¶[0581], [1135]–[1136] (discussing the need for connections to manage unintentional disconnection), [1153]; EX1007, 1:16–32, 1:54–2:11, 2:43–47 (discussing importance of maintaining connections and handling disruptions); *supra* Section IX.A.1.b.1.) Thus, a POSITA would have had reason to consider *Berg* when contemplating and implementing the teachings of *Wookey*. (EX1002 ¶146.) And when collectively considered, such a POSITA would have been motivated to modify *Wookey*’s **client machine 10** to use a protocol like *Berg*’s RUDP in order to set up an RUDP connection with **remote machine 30’**. (*Id.*)

As discussed, *Wookey* describes using “code” to cause **client machine 10** to use the network application (browser 280) to establish a second connection between **client machine 10** and **remote machine 30’** using a protocol that can be different

Petition for *Inter Partes* Review
Patent No. 10,075,564

from the one used to connect **client machine 10** and **remote machine 30**. *Wookey* further discloses the types of characteristics a POSITA would have considered concerning the protocol used for the second connection, such as negotiating parameters related to the connection, reducing unintentional termination due to an imperfect connection, detecting and handling disconnections, and specifying a permissible inactive time before connection termination. (*See supra* Section IX.A.1.b.1.) A POSITA would have been thus motivated by *Wookey*'s disclosures to consider protocols that would provide the desired features to *Wookey* for the second connection with **remote machine 30'**, including the features associated with the RUDP described by *Berg*. A POSITA would have been motivated to incorporate a protocol like *Berg*'s RUDP (which is a non-TCP protocol) in *Wookey*'s use of the "code" and "network application" for establishing a connection between **client machine 10** and **remote machine 30'**. (EX1002 ¶147.)

A POSITA would have also recognized the benefits of using the RUDP protocol described in *Berg* with *Wookey*'s system/method, such as providing **client machine 10** with an improved reliable connection to **remote machine 30'** with the versatility of negotiated timeout parameters consistent with *Wookey*'s desired characteristics for the second connection. *See KSR*, 550 U.S. 416–17. Indeed, by using a protocol like *Berg*'s RUDP protocol, *Wookey*'s method would enable

Petition for *Inter Partes* Review
Patent No. 10,075,564

negotiation of properties like the timeout values on a per-connection basis over a reliable connection (EX1007, 24:40–47) and provide alerts to the nodes if the session fails (*id.*, 2:55–60). (EX1002 ¶148.) *Wookey*’s method for “providing a client with a reliable connection to a host service” (EX1005 ¶[0077]) would therefore have been improved by providing a connection that is “very flexible ... [and] suitable for a variety of transport uses,” as provided by *Berg* (EX1007, 17:9–16, 17:42–46.). (EX1002 ¶148.)

It would have been a foreseeable and straightforward implementation to configure *Wookey*’s **client machine 10** such that the “code” (e.g., the encoded URLs in HTML page 288 or the HTML page 288 including such URLs), when used, would cause **client machine 10** (e.g., by way of its web browser 280) to operate in accordance with *Berg*’s RUDP (“non-TCP protocol”). (EX1002 ¶149.) See *KSR*, 550 U.S. at 416. For example, as noted above, *Wookey* discloses that “*the URL includes a file, or a reference to a file, that contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (EX1005 ¶[0216].) In light of the reasons discussed above, a POSITA would have been motivated to configure *Wookey*’s encoded URL such that it includes information resulting in **client machine 10** utilizing the web browser application 280 to operate in accordance with a non-TCP protocol (like *Berg*’s

Petition for *Inter Partes* Review
Patent No. 10,075,564

RUDP) to initiate a process of establishing a second connection with **remote machine 30'**. (EX1002 ¶149.) A POSITA would have achieved this implementation by combining well-known technologies using known methods, such as known network design concepts and technologies described above by *Wookey* and *Berg*, and known in the art at the time. (*Id.*)

Berg's protocol is flexible and suited for various uses and allows the nodes to negotiate properties like the timeout values on a per-connection basis over a reliable connection. (EX1007, 24:40–47.) These are features that *Wookey* itself recognizes (EX1005 ¶[0721]), and are features that would have improved *Wookey*'s process just as described in *Berg*. Accordingly, such a combination would have yielded the predictable result of providing a network communication exchange that minimizes disruptions caused by timeout issues. (EX1002 ¶150); *see KSR*, 550 U.S. at 417.

The *Wookey-Berg* combination would have involved the use of known technologies (e.g., aspects of similar protocols) and design concepts and processes to obtain the foreseeable result of a reliable connection between *Wookey*'s **client machine 10** and **remote machine 30'**. (EX1002 ¶151.) Indeed, the above-modification would have involved the substitution of features from one reliable protocol from a finite number of available alternative reliable communication protocols, such as that described by *Berg*. (*Id.*) Thus, a POSITA would have had a

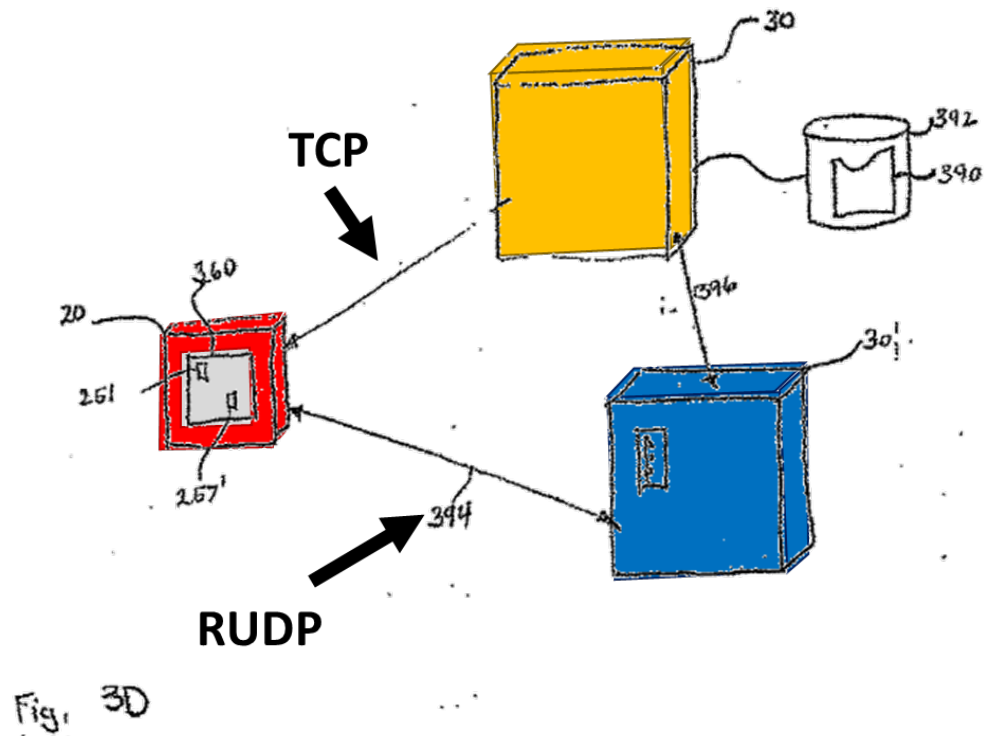
Petition for *Inter Partes* Review
Patent No. 10,075,564

reasonable expectation of success in the above-modification. (*Id.*) See *Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1364 (Fed. Cir. 2007) (“only a reasonable expectation of success, not a guarantee, is needed” in an obviousness analysis).¹⁴

A POSITA would have been skilled and knowledgeable to configure the above modification in various ways, while taking into account any known programming, design, and other related concepts, limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. (EX1002 ¶152.) For example, a POSITA would have been motivated based on such disclosures to configure *Wookey*’s system and process consistent with the non-limiting example reflected below.¹⁵

¹⁴ There is no evidence of nonobviousness at this time that would overcome the overwhelming evidence of obviousness as presented herein. Petitioner reserves the right to address any such evidence/argument should PO later present any in this proceeding.

¹⁵ The configuration discussed below is exemplary and not limiting. (EX1002 ¶152.)



(EX1005, FIG. 3D (annotated); EX1002 ¶152.)

* * *

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 24.b(1). (EX1002 ¶153; *see also infra* Section IX.A.1.b(2), IX.A.1.c–f.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

- (2) [... a client node] including one or more processors in communication with a non-transitory memory storing a network application that is configured to operate in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, []

The *Wookey-Berg* combination discloses and/or suggests and renders obvious limitation 24.b(2). (EX1002 ¶¶154–164.) As discussed below, the **client machine 10** (“client node”) in the *Wookey-Berg* combination includes a CPU 102 (“one or more processors”) in communication with a main memory unit 104 and cache memory 140 (individually or collectively, “non-transitory memory”) storing a web browser application 280 (“network application”) that is configured to operate in accordance with the RUDP protocol (“a non-transmission control protocol (TCP)”) that would have operated above an Internet Protocol (IP) layer and below a Hypertext Transfer Protocol (HTTP) application layer, as claimed.

For example, *Wookey* explains that **client machine 10** is a *computer* (EX1005 ¶[0160]), with architecture like shown in Figures 1A and 1B (*id.* ¶[0021]) (below), that includes the CPU 102 in communication with main memory 104 and

Petition for *Inter Partes* Review
 Patent No. 10,075,564

cache memory 140, which are non-transitory memory devices as explained above.
 (EX1005 ¶¶[0162]–[0165]; *supra* Section IX.A.1.a; EX1002 ¶155.)¹⁶

FIG. 1A

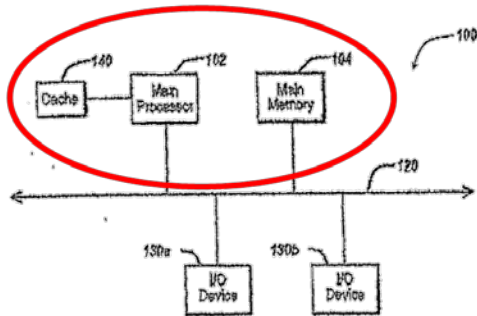
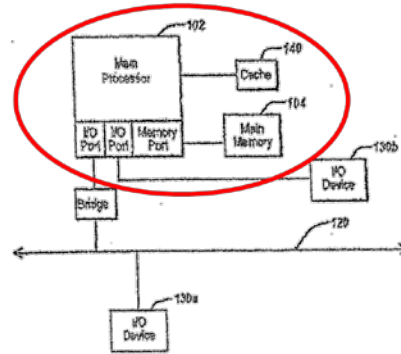


FIG. 1B



(EX1005, FIGS. 1A–1B (annotated); EX1002 ¶¶155–156.)

Wookey further discloses that the memory in the computer is used to store the web browser application 280 (“network application”). For example, **client machine 10** uses its memory to store and its “display presentation program” (web browser application). (EX1005 ¶[0171]; *see also id.* ¶¶[0159], [0171] (explaining machines 10 can include persistent or volatile memory for storing application programs), [0192], [0538]; EX1002 ¶157.)

¹⁶ *Wookey* uses the terms “Main Processor 102,” “microprocessor 102,” “processor 102,” and “central processing unit (CPU) 102” interchangeably. (EX1005 ¶¶[0161], [0163]–[0164], FIGS. 1A–B; EX1002 ¶155 n.15.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

For example, *Wookey* explains that **client machine 10** “can access a resource ... displayed in the Resource Neighborhood web page” (EX1005 ¶[0216]), which is executed via the web browser application 280 (*id.* ¶[0192]) over the World Wide Web. *Wookey*’s web browser application 280 initiates requests and accesses resources according to HTTP (EX1005 ¶[0155]), which was a known communication protocol at the application layer of an OSI model or TCP/IP model¹⁷ for the World Wide Web. (EX1002 ¶¶158–159; EX1018, 7–8, 12.)

Indeed, *Wookey* provides that the communications link between the remote and client machines may use an “application layer protocol, such as the Hypertext Transfer Protocol (HTTP).” (EX1005 ¶[0155].) Further, **client machine 10** includes an “HTTP client agent,” such as a web browser, which “can use any type of protocol.” (*Id.* ¶[0159]; EX1002 ¶159.)

Thus the RUDP in the *Wookey-Berg* combination would have operated below the application layer on which the HTTP operates. (EX1002 ¶160.) This is because, in the *Wookey-Berg* combination, the RUDP would have run on top of the User

¹⁷ For purposes of this IPR, regardless of the model (OSI or TCP/IP) considered at the time, a POSITA would have had the same understanding as explained herein. (EX1002 ¶159 n.16.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

Datagram Protocol (UDP) protocol software 204 at a transport layer, and information exchanged via the RUDP between the client and server would have been communicated to the application layer where the HTTP-based web browser operated. (*Id.*) The transport layer is below the application layer. (*Id.*) Similarly, the RUDP in the *Wookey-Berg* combination would have operated above an IP layer because the transport layer operates above the IP layer. (*Id.*)

Berg confirms this understanding. Specifically, *Berg* discusses the logical layers that computer systems use to communicate over a network using RUDP. (EX1007, 8:3–8; *see supra* Section IX.A.1.b.1.a.) As shown below, the **RUDP layer 206** operates above **IP layer 202** and below **Signaling Protocols Application(s) layer 208** (an application layer, where the HTTP would operate, i.e., “a hypertext transfer protocol (HTTP) application layer”). (EX1007, FIG. 2, 8:3–18, 8:24–31; *see also id.*, 1:15–59, 5:27–55.)

Petition for *Inter Partes* Review
 Patent No. 10,075,564

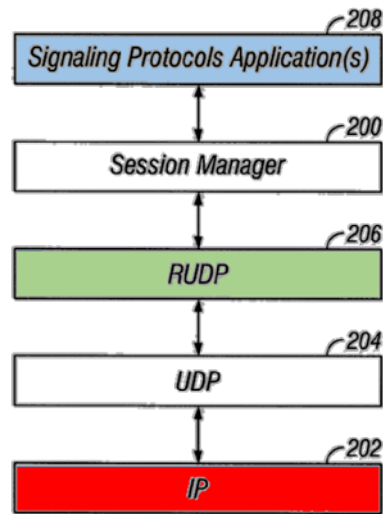


FIG. 2

(EX1007, FIG. 2 (annotated); EX1002 ¶161.)

Thus, the combination of *Wookey-Berg* discloses and/or suggests and renders obvious a **client machine 10** “including one or more processors in communication with a non-transitory memory storing a network application that is configured to operate in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer,” as claimed in limitation 24.b. (EX1002 ¶¶162–163.)

* * *

Accordingly, the *Wookey-Berg* combination discloses and/or suggests limitation 24.b. (EX1002 ¶164.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

- c) **receive, from a server node, a non-TCP packet during a setup of a non-TCP connection;**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶165–170.) As explained below, the “code” in the combined *Wookey-Berg* system/method would, when used by **client machine 10**, results in **client machine 10** utilizing the network browser 280 to operate in accordance with the RUDP protocol to receive, from **remote machine 30’** (“server node”), a SYN message (“non-TCP packet”) during a setup of the RUDP connection (“non-TCP connection”). (*Id.*)

For example, *Berg* explains that RUDP allows characteristics of each connection to be individually configured so that protocols with different transport requirements can be simultaneously implemented. (EX1007, 17:42–46.) Thus, when a client initiates a connection, it sends a SYN segment that contains the negotiable parameters defined by the client’s Upper Layer Protocol (ULP). (*Id.*, 24:40–42; *see also id.*, 18:56–62.) Upon receiving the client’s SYN message, the server determines to either “accept” the proposed parameters in the client’s SYN message or propose different parameters. (*Id.*, 24:42–44.) In either case, the server returns the parameters in its SYN response to the client. (*Id.*) The client receives, from the server, a SYN message (“non-TCP packet”) that contains proposed parameters during the initiation of the RUDP connection (“non-TCP connection”).

Petition for *Inter Partes* Review
Patent No. 10,075,564

(*Id.*, 24:44–47.) The client can then choose to accept the server’s parameters by responding with an ACK message to establish the connection or send a reset (RST) segment to the server to refuse the connection. (*Id.*, 24:44–47; EX1002 ¶167.)

The RUDP messages sent to set up the RUDP connection, including the SYN messages, are UDP packets that include an RUDP header. (EX1007, 17:59–18:20; *see also id.*, 19:1–40.) RUDP messages are non-TCP packets because they are communicated using RUDP, which is a non-TCP protocol. (*Supra* Section IX.A.1.b.1.a.) Therefore, the SYN response sent from **remote machine 30’** and received by **client machine 10** in the *Wookey-Berg* combination is a non-TCP packet. (EX1002 ¶168; *supra* Section VIII.) Furthermore, the connection established according to an RUDP protocol is a non-TCP connection. (EX1002 ¶168; *supra* Section IX.A.1.b.1.a.)

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 24.c. (EX1002 ¶¶169–170.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

- d) **identify metadata, that specifies a number of seconds or minutes, in an idle time period parameter field in the non-TCP packet, for an idle time period, where, as a result of a detection of the idle time period, the non-TCP connection is subject to deactivation; and**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶171–194.) The combined *Wookey-Berg* system/method discloses and/or suggests that **client machine 10** identifies various negotiable parameters (“metadata”) in the timeout parameter field in the SYN segment (“non-TCP packet”), including a timeout value that specifies a number of seconds for an idle time period, where, as a result of a detection of the timeout value, the RUDP connection is subject to deactivation. (*Id.*; *see supra* Section IX.A.1.c.)

Berg discloses identifying the negotiable parameters in the SYN segment (“non-TCP packet”) for establishing an RUDP connection. For example, *Berg* explains that when the client receives the SYN segment sent from the server, it can *accept* or *reject* the proposed parameters (such as the null segment timeout value) in the received SYN segment. (EX1007, 24:44–48.) Thus, the client necessarily identifies each parameter in the SYN segment to decide whether it will accept or reject the proposed parameters. (EX1002 ¶173.) The *Wookey-Berg* combination would have thus been configured such that **client machine 10** identifies data

Petition for *Inter Partes* Review
Patent No. 10,075,564

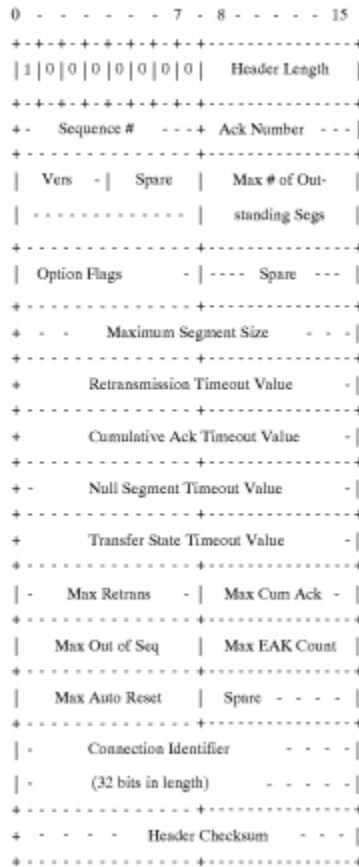
(“metadata”) in the parameter fields in the RUDP SYN segment (“packet”). (EX1002 ¶174.)

To the extent the *Wookey-Berg* does not disclose the claimed “identify[ing],” it would have been obvious to modify the functionality in the combined *Wookey-Berg* system/process such that it identifies the negotiable parameters as claimed. (EX1002 ¶175.) A POSITA would have been motivated to configure the *Wookey-Berg* system/process to include functionality that identifies the parameters in the SYN segment to facilitate the subsequent process of accepting or rejecting the parameters in a manner consistent with the features described by *Berg*. (EX1007, 24:44–48; EX1002 ¶176.)

For example, Figure 2, below, of the RUDP specification,¹⁸ illustrates the SYN segment with all of its negotiable parameters for the RUDP connection:

¹⁸ The RUDP specification is included in *Berg* as “Appendix 1.” (EX1007, 16:59–25:54; *see also* EX1011, 6.)

Petition for *Inter Partes* Review
Patent No. 10,075,564



(EX1007, 19:1–40; EX1002 ¶177.)

The *Wookey-Berg* combination would have thus been configured such that **client machine 10** identifies data (“metadata”) in the parameter fields in the RUDP SYN segment (“non-TCP packet”). (EX1002 ¶178.)

As explained below, the negotiable parameters for (1) the null segment timeout value and (2) transfer state timeout value each separately represent “metadata, that specifies a number of seconds or minutes, in an idle time period parameter field in the non-TCP packet, for an idle time period, where, as a result of

Petition for *Inter Partes* Review
Patent No. 10,075,564

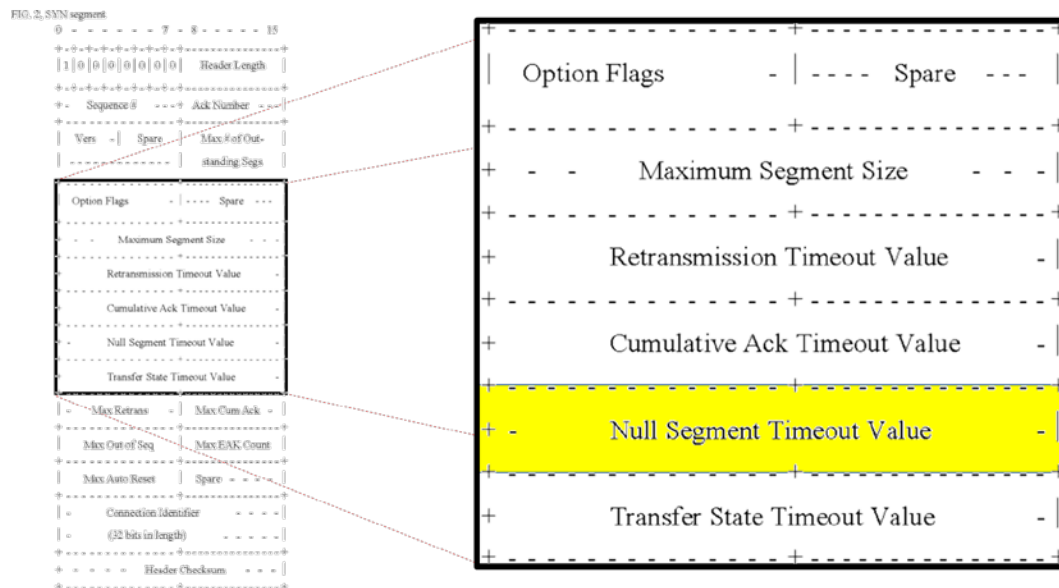
a detection of the idle time period, the non-TCP connection is subject to deactivation,” as claimed. (EX1002 ¶179.)

(1) Null Segment Timeout Value

Berg discloses that a null segment timeout value (“metadata”), specified in millisecond (“a number of seconds,” as claimed), is identified from a null segment timeout value parameter field (“idle time period parameter field”) in an RUDP SYN segment (“non-TCP packet”). (*E.g.*, EX1007, 18:58–60, 19:22–25, 20:36–42, 22:25–33; *see also* EX1002 ¶180.) The null segment timeout value indicates the amount of time to wait before sending a null segment when the connection has been idle (e.g., when a data segment has not been sent or received). (EX1007, 20:36–41.) The null segment is sent to determine “if the other side of a connection is still active.” (*Id.*, 22:25–33, 20:36–41.) Thus this period of waiting by the client (when the connection is idle) before the client sends a null segment to the server to verify whether the connection is active represents an “idle time period,” as claimed.

Petition for *Inter Partes* Review
 Patent No. 10,075,564

(EX1002 ¶180.) Figure 2 of the RUDP specification illustrates the null segment timeout value parameter field that specifies the **null segment timeout value**:



(*Id.*, 19:1–40 (excerpted and annotated); *see also* EX1002 ¶180.)

Furthermore, as a result of detecting a null segment timer’s expiration, which is set based on the null segment timeout value, the RUDP connection is “subject to deactivation,” as claimed. (EX1002 ¶181.) For example, at the client-side, the null segment timeout value is used to set a null segment timer “for sending a null segment *if a data segment has not been sent.*” (EX1007, 20:36–42, 23:45–49, 22:25–32.)¹⁹ As a result of detecting the null segment timer’s expiration,

¹⁹ *Berg* uses the terms “null segment” and “NUL segment” interchangeably. (EX1007, 20:36–42, 22:25–32; EX1002 ¶181.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

a null segment is sent to the other side to determine if the connection is still valid. (*Id.*, 23:45–59.) When “a NUL segment is received, an RUDP implementation *must* acknowledge the segment if a valid connection exists.” (*Id.*, 22:27–30.) On the other hand, if the null segment is not received or acknowledged, the connection is subject to being closed via an “auto reset.” (*Id.*, 23:45–59; EX1002 ¶181.)

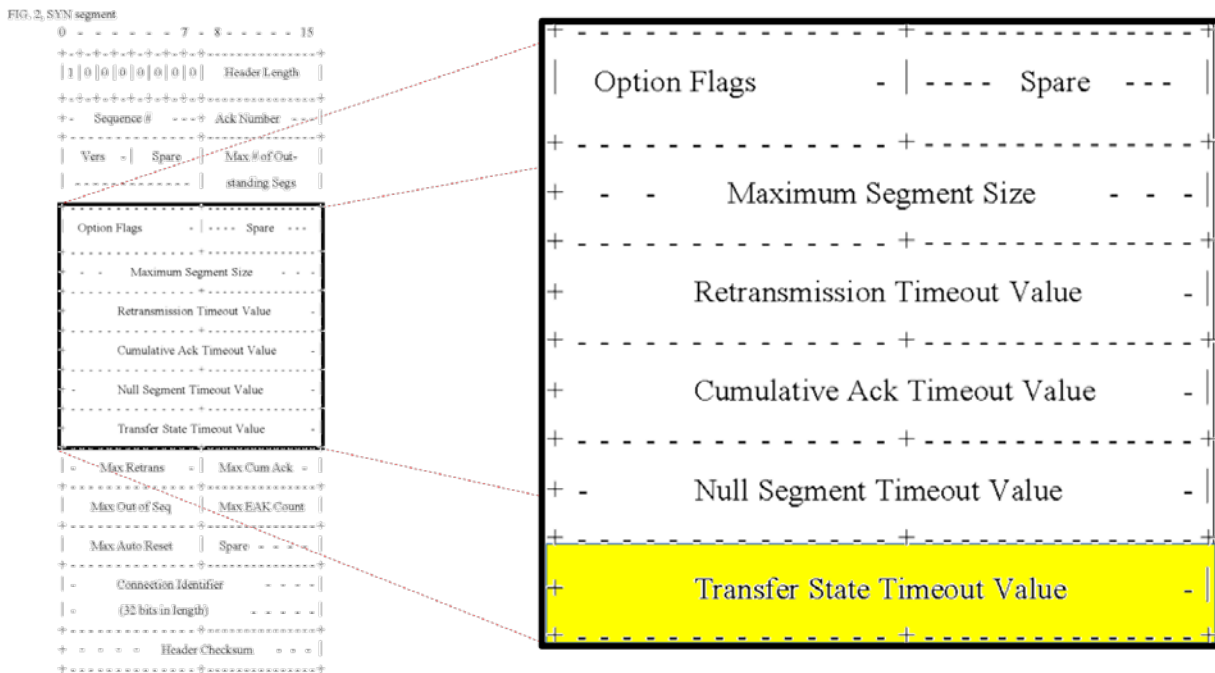
The “auto reset,” which is triggered upon the null segment timer’s expiration, is a form of deactivation of the RUDP connection. (EX1002 ¶182; *see also* EX1007, 21:15–23, 23:60–24:4.) For example, an “auto reset” signals a problem with the connection (e.g., connection failure) and that the host and client need to deactivate the connection. (EX1007, 23:60–24:4; *see also id.*, 21:15–23, 24:15–25.) When an auto-reset is necessary, either side of the connection can send a reset (RST) segment to close or reset a connection. (*Id.*, 23:60–24:4; *see also id.*, 18:27–28, 22:20–24.) Either closing or resetting a connection is a form of deactivation for the connection. (EX1002 ¶¶182–184.)

Thus the client identifies the null segment timeout value (“metadata”), specified in milliseconds (“a number of seconds,”) in a timeout parameter field in the SYN segment (“non-TCP packet”) for an idle time period, where, as a result of the detection of the idle time period (via the expiration of the null-segment timer

which utilizes this value), the RUDP (“non-TCP”) connection is subject to deactivation. (EX1002 ¶185.)

(2) Transfer State Timeout Value

Berg also discloses that a transfer state timeout value (“metadata”), which is specified in a number of seconds, is identified from a transfer state timeout value parameter field (“idle time period parameter field”) in an RUDP SYN segment (“non-TCP packet”). (*E.g.*, EX1007, 19:25–29, 20:36–42, 22:25–33; *see also* EX1002 ¶186.) Figure 2, below, of the RUDP specification, illustrates the transfer state timeout value parameter field that specifies the **transfer state timeout value**:



(EX1007, 19:1–40 (excerpted and annotated); EX1002 ¶186.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

When an RUDP connection fails, the nodes will save the current connection state and close the current RUDP connection. (EX1007, 24:15–25.) The nodes will then wait to transfer the saved state of the previous failed RUDP connection to another connection using a transfer state timer (set to the transfer state timeout value). (*Id.*; *see also id.*, 20:42–47.) When the transfer state timer expires, the connection’s data is freed up, and thus the RUDP connection is subject to deactivation. (*Id.*, 24:15–25.) The transfer state timeout value is how long either node will wait before deeming the connection state lost. (*Id.*) Thus, the transfer state timer before it expires causes the state associated with the failed RUDP connection to be kept at least partially active while waiting to transfer to another connection. (EX1002 ¶¶187–188.) Once the time period expires, the saved connection’s data is removed. (EX1007, 24:15–25.) Because the connection’s data is no longer saved and the connection cannot be transferred to a new connection upon expiration of the transfer state timer, detection of such expiration subjects the connection to deactivation. (*Id.*; *see also id.*, 20:42–47) This period of waiting (when the connection is idle) before the RUDP connection is deemed inactive represents an “idle time period.” (EX1002 ¶189.)

Thus, the client identifies the transfer state timeout value (“metadata”), specified in a number of seconds, in a timeout parameter field in the SYN segment

Petition for *Inter Partes* Review
Patent No. 10,075,564

(“non-TCP packet”) for an idle time period, where, as a result of the detection of the idle time period (via the expiration of the transfer state timer that utilizes this timeout value), the RUDP (“non-TCP”) connection is subject to deactivation. (EX1002 ¶¶190–191.)

* * *

This understanding that timeout values are metadata is confirmed by the ’564 patent, which explains that the metadata can be a value indicating a “duration of time” (EX1001, 20:58–64), which is what the timeout values in *Berg* contain (EX1007, 20:36–47; EX1002 ¶192.)

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been further configured to include the process of **client machine 10** identifying the null segment timeout value or transfer state timeout value (“metadata”), that specifies a number of seconds or minutes, in the null segment timeout value parameter field or the transfer state timeout value parameter field (“idle time period parameter field”) in the RUDP SYN segment (“non-TCP packet”), for an idle time period, where, as a result of a detection of the idle time period, the RUDP (“non-TCP”) connection is subject to deactivation, as claimed. (EX1002 ¶¶193–194; *see supra* Section IX.A.1.b.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

- e) **determine, based on the metadata, a timeout attribute associated with the non-TCP connection;**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶195–200.) **Client machine 10** in the *Wookey-Berg* combination determines timeout attributes, such as a null segment timer and transfer state timer, based on the parameters (“metadata”)—e.g., the null segment timeout and transfer state timeout values—associated with the RUDP connection (“non-TCP connection”). (*Id.*; *Supra* Section IX.A.1.d.)

For example, *Berg* explains that the client determines the timeout value for a null segment timer (“timeout attribute”) associated with the established RUDP connection (“non-TCP connection”) based on the null segment timeout value (“metadata”). (EX1007, 20:36–38, 23:46–59.) The client’s null segment timer is set based on the null segment timeout value (e.g., when the client accepts the null segment timeout value proposed by the server). (*Id.*, 23:45–59; *see also id.*, 20:36–41; EX1002 ¶197; *supra* Section IX.A.1.d.1.) *See, e.g., Unwired Planet, LLC*, 841 F.3d at 1002; *Power Integrations, Inc.*, 843 F.3d at 1336-37. Accordingly, **client machine 10** determines, based on the null segment timeout value (“metadata”), a null segment timer (“timeout attribute”) associated with the RUDP (“non-TCP”) connection. (EX1002 ¶197; *see also* Sections IX.A.1.b, IX.A.1.d.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

Berg also explains that the client determines the timeout value for a transfer state timer (“timeout attribute”) associated with the established RUDP connection (“non-TCP connection”) based on the transfer state timeout value (“metadata”). (EX1007, 20:42–45, 23:60–61, 24:21–25; EX1002 ¶198; *supra* Section IX.A.1.d.2.) The client maintains a “transfer state timer” for the RUDP connection with a transfer state timeout value. (EX1007, 24:21–23; *see also id.*, 20:42–47.) Thus, in the *Wookey-Berg* combination, **client machine 10** determines, based on the transfer state timeout value (“metadata”), a transfer state timer (“timeout attribute”) associated with the RUDP “(non-TCP)” connection. (EX1002 ¶198; *see also* Sections IX.A.1.b, IX.A.1.d.)

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been further configured to include the process of **client machine 10** determining the timeout attributes, such as a null segment timer and transfer state timer, based on the parameters (“metadata”)—e.g., the null segment timeout and transfer state timeout values—associated with the RUDP connection (“non-TCP connection”). (*See supra* Section IX.A.1.b; EX1002 ¶199.)

* * *

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 24.e. (EX1002 ¶200.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

- f) **wherein the code, when used by the client node, results in the client node utilizing the network application to operate in accordance with the TCP protocol to perform a three-way TCP handshake with another server node for setting up a TCP connection with the another server node that is different than the non-TCP connection.**²⁰

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶201–211.) For example, *Wookey* discloses that a first **remote machine 30** (“another server node”) causes code (e.g., (1) one of the encoded URLs in the received HTML page 288, or (2) the HTML page 288, which includes the encoded URLs) to be sent to a **client machine 10** (“client node”) over a TCP/IP connection. (*Supra* Section IX.A.1.b.) The code causes **client machine 10** to access various resources by communicating with remote machines when **client machine 10** uses the code. (*Supra* Section IX.A.1.b.) As explained below, the code may identify that a resource is located on **remote machine 30** (“another server node”), e.g., while one encoded URL on the HTML page may identify a resource

²⁰ The ’564 patent specification does not provide any disclosure describing the claimed “the code” features. (EX1002 ¶201 n.18.) Therefore, while Petitioner addresses this limitation with respect to the asserted prior art, Petitioner does not concede that the ’564 patent provides adequate disclosures of such claimed features.

Petition for *Inter Partes* Review
Patent No. 10,075,564

located on **remote machine 30'** ("server node") as discussed above for limitations 24.b–c, another encoded URL in the HTML page may identify a resource located on **remote machine 30** ("another server node"). (*See supra* Sections IX.A.1.b–c.) So when **client machine 10** selects an encoded URL on the HTML page identifying **remote machine 30**, a new connection may be established between **remote machine 30** ("another server node") and **client machine 10** ("client node") as discussed below. (EX1002 ¶202.)

Regarding the connection, *Wookey* provides that **remote machine 30** presents an enumeration of resources available to the user on **client machine 10**. (*Supra* Section IX.A.1.b; EX1005 ¶¶[0207], [0213], [0625], [0673].) For example, "remote machine 30 may create a page describing a display of resources, hosted by a plurality of machines" (EX1005 ¶[0625]) available to **client machine 10**. (*See also id.* ¶[0213].) The page is displayed on **client machine 10**, and the user on the machine can make "a request to access one of the hosted resources." (*id.* ¶[0625]). (*See also id.* ¶¶[0213], [0216].) *Wookey* explains that "the first remote machine 30 [may] host[] the selected one of the available resources." (*Id.* ¶[0627]; *see also id.* ¶¶[0006], [0673], [0707].) Thus *Wookey* discloses a scenario where **remote machine 30** can host the selected resource for **client machine 10**. (EX1002 ¶203.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

In such a scenario, “remote machine 30 makes the resource available to the user *over a new connection*.” (EX1005 ¶[0627].) So when **client machine 10** selects a resource by clicking on an icon with an associated encoded URL on the HTML page 288 specifying **remote machine 30** (where both the encoded URLs on the HTML page 288 and the HTML page which includes the encoded URLs are “code”), **client machine 10** utilizes the web browser application 280 (“network application”) to establish a new connection with **remote machine 30** to access the selected resource. (EX1002 ¶204; EX1005 ¶[0216].)

The new connection between **remote machine 30** and **client machine 10** “can be established using a variety of communication protocols” (EX1005 ¶[0156]) including “TCP/IP” (*id.*).²¹ Thus *Wookey* discloses that **client machine 10** may

²¹ As explained, *Wookey* discloses that **client machine 10** can establish a second connection with the selected resource (e.g., **remote machine 30** or **remote machine 30'**) using “any type of protocol” including both TCP and non-TCP protocols. (EX1005 ¶¶[0155], [0159], [0216], [0225]; *supra* Section IX.A.1.b.) Thus, *Wookey* discloses that the encoded URLs on the HTML page pointing to the various resources can be used to establish connections using different protocols (e.g., one using TCP and another using non-TCP). (EX1002 ¶205 n.19.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

establish a second TCP connection with **remote machine 30** using a TCP protocol. (EX1002 ¶¶205–206.)

A three-way handshake method is necessarily performed to establish a TCP connection. (EX1002 ¶207.) Indeed, RFC 793, the TCP specification²², provides that a three-way handshake is necessary (EX1008, 32) and that it is used to establish a TCP connection (*id.*, 34.) (*See also id.*, 31–32, 34–37; EX1002 ¶207.) The ’564 patent acknowledges that the three-way handshake is required to establish a TCP connection, citing RFC 793. (EX1001, 13:49–63; EX1002 ¶208.)

Thus, to establish a TCP connection between **client machine 10** and **remote machine 30**, **client machine 10** (“client node”) utilizes the web browser 280 (“network application”) to operate in accordance with the TCP protocol to perform a three-way TCP handshake with **remote machine 30** (“another server node”) for setting up a TCP connection with **remote machine 30**, as claimed. (EX1002 ¶209.) Accordingly, the second TCP connection between **client machine 10** and **remote machine 30** (“another server node”) in the *Wookey-Berg* combination is different

²² RFC 793 is incorporated by reference in the ’564 patent (EX1001, 1:48–56, 10:32–33). (EX1002 ¶207 n.20.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

than the RUDP connection (“non-TCP connection”) between **client machine 10** and **remote machine 30’**. (*Id.*; see also *supra* Section IX.A.1.b.)

Therefore, the *Wookey-Berg* combination discloses and/or suggests limitation 24.f. (EX1002 ¶¶210–211.)

2. Claim 25

- a) **A system including the another server node which includes the non-transitory computer readable medium of claim 24,**

To the extent limiting, the *Wookey-Berg* combination discloses and/or suggests and renders obvious the limitations of this preamble for at least the same reasons discussed above for claim 24. (*Supra* Section IX.A.1; EX1002 ¶¶212–214.) For example, *Wookey* provides an environment (“system”) (Figure 1) including **client machine 10**, **remote machine 30’** (“server node”), and **remote machine 30** (“another server node”). (EX1005 ¶[0020]; *see also id.* ¶¶[0002] (“systems”), [0016], [0196].) Further as discussed, **remote machine 30** (“another server node”) includes the non-transitory computer readable medium (e.g., main memory unit 104 and/or cache memory 140). (*Supra* Sections IX.A.1.a–b; EX1002 ¶¶213–214.)

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 25.a. (EX1002 ¶214; *see also infra* Section IX.A.2.b.)

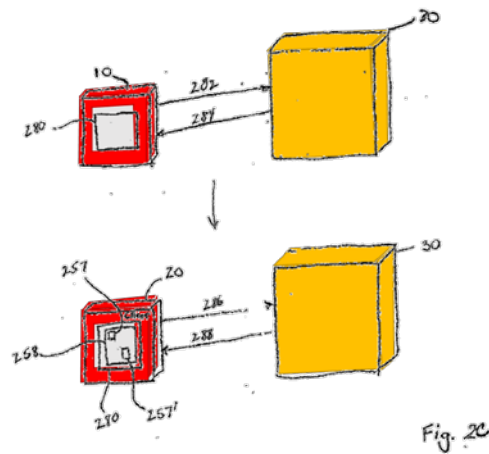
- b) **wherein the system is configured to communicate the code to the client node utilizing the TCP connection, for setting up the non-TCP connection with the server node.**²³

Claim 25.b requires the system to communicate code to a client node via a TCP connection for setting up the non-TCP connection with the server node. (EX1002 ¶¶215–218.) As explained above for claim 24, **remote machine 30** in the *Wookey-Berg* combination is configured to communicate the “code” (e.g., (1) one of the encoded URLs in the received HTML page 288, or (2) the HTML page 288, which includes the encoded URLs) to **client machine 10** utilizing the TCP connection, for setting up the RUDP (“non-TCP”) connection with **remote machine 30’** (“server node”). (*Supra* Sections IX.A.1.b, IX.A.1.f; EX1002 ¶217.) An exemplary annotated Figure 2C below from *Wookey* illustrates the communication of HTML page (e.g., page 288), including code for establishing an RUDP connection, from **remote machine 30** to **client machine 10** in the combined *Wookey-Berg* system.²⁴

²³ See n.20.

²⁴ See n.6.

Petition for *Inter Partes* Review
Patent No. 10,075,564



(EX1005, FIG. 2C (annotated); EX1002 ¶217.)

For these reasons, the *Wookey-Berg* combination discloses and/or suggests limitation 25.b. (EX1002 ¶218.)

3. Claim 26

- a) **The system of claim 25, wherein the system is configured to communicate the code to the client node via the TCP connection utilizing a hypertext transfer protocol (HTTP).**²⁵

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for at least the same reasons discussed above for limitations 24.b.1–2, 24.f, and 25.a–b. (*Supra* Sections IX.A.1.b.1–2, IX.A.1.f, IX.A.2.a–b; EX1002 ¶¶219–224.) For example, as discussed above, **remote machine 30** in the combined *Wookey-Berg* system is configured to communicate the code to **client machine 10**, via a HTTP-based web browser application 280, utilizing the TCP connection. (*Supra* Sections IX.A.1.b.1–2, IX.A.1.f, IX.A.2.a–b; *see also* EX1005 ¶¶[0155] (“The communications link 150 may use ... any application layer protocol, such as the Hypertext Transfer Protocol (HTTP).”), [0552] (the client and remote machine “may communicate using a protocol, such as http”).)²⁶ Thus, the TCP connection established between **client machine 10** and the **machine 30** permits the

²⁵ *See* n.20.

²⁶ **Client machine 10** and **remote machine 30** as shown in Figures 2C and 3D may operate in a similar manner to the configuration shown in Figure 15B. (EX1002 ¶222 n.24; EX1005 ¶¶[0549]–[0558], FIG. 15B.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

system to communicate the code (e.g., the encoded URLs in the HTML page 288 or the HTML page 288 itself) via HTTP-based applications (e.g., the web server and web browser application 280). (EX1002 ¶222; *see also supra* Section IX.A.1.b.2.)

For these reasons, the *Wookey-Berg* combination discloses and/or suggests limitation 26.a. (EX1002 ¶¶223–224.)

4. Claim 27

- a) **The system of claim 26, wherein the code, when used by the client node, results in the client node operating such that the non-TCP connection is setup between the client node and the server node instead of the TCP connection in order to permit communication, between the client node and the server node, of the timeout attribute,**²⁷

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶225–228.) As described above for claims 24–26, **remote machine 30** in the combined *Wookey-Berg* system causes the “code” (e.g., (1) one of the encoded URLs in the HTML page 288, or (2) the HTML page 288, which includes the encoded URLs) to be sent to **client machine 10** (“client node”) via a TCP connection. (*Supra* Sections IX.A.1.b, IX.A.2–3.) As explained above, this code, when used by **client machine 10** (“client node”), results in **client machine 10** operating such that the RUDP (“non-TCP”) connection is set up between **client machine 10** and **remote machine 30’** (“server node”) instead of the TCP protocol to permit communication. (*Supra* Sections IX.A.1.b–f, IX.A.2–3; *see also supra* EX1002 ¶¶226–227.) During the setup of the RUDP connection, **client machine 10** and **remote machine 30’** in the combined *Wookey-Berg* system negotiate timeout

²⁷ See n.7. (EX1002 ¶225 n.25.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

values, e.g., the null segment timeout value and transfer state timer value, which are used to set up the timeout attributes for the RUDP connection. (*Supra* Sections IX.A.1.c–f; EX1002 ¶227.) In other words, the communication of these timeout values is not performed over a TCP connection but instead over the RUDP connection.

Accordingly, the *Wookey-Berg* combination discloses and/or suggests limitation 27.a. (EX1002 ¶228.)

- b) **where the timeout attribute is not communicated when setting up the TCP connection between the system and the client node, but is communicated when establishing the non-TCP connection between the client and the server node.**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶229–232.) As explained above, in the *Wookey-Berg* combination, the timeout attributes—e.g., the null segment timer and transfer state timer—are values associated with its RUDP connection with **remote machine 30'** (“server node”). (*Supra* Sections IX.A.1.b–f, IX.A.2–3.) The timeout values associated with these RUDP-based timers are communicated when establishing the RUDP (“non-TCP”) connection between **remote machine 30'** (“server node”) and **client machine 10** (“client node”), and not when setting up a TCP connection.

Petition for *Inter Partes* Review
Patent No. 10,075,564

(*Supra* Sections IX.A.1.b–f, IX.A.2–3; *see also* EX1007, 18:26–60, 24:40–47; EX1002 ¶230.)

Additionally, these timeout values associated with the timeout attributes are not communicated when setting up the TCP connection (e.g., by way of a three-way handshake) between “the system and the client node” (e.g., between **remote machine 30** and **client machine 10**), but rather are communicated via an RUDP SYN segment (a non-TCP packet) when establishing the RUDP (“non-TCP connection”) between **client machine 10** (“client node”) and **remote machine 30** (“server node”). (*Supra* Sections IX.A.1.b–f, IX.A.2–3; *see also* EX1007, 18:56–21:35, 24:40–47; EX1002 ¶231.) Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 27.b. (EX1002 ¶232.)

5. Claim 28**a) A computer-implemented method, comprising:**

To the extent limiting, *Wookey* discloses this limitation for at least the same reasons as above for limitations 24a–f, explaining *Wookey*’s disclosure of a computer-readable medium including code for performing steps similar to those recited in claim 28. (*See supra* Sections IX.A.1.a–f; EX1002 ¶¶233–234; *infra* Sections IX.A.5.b–e; EX1005 ¶¶[0002], [0285] (“method”), Abstract (“method”).)

b) Claim 28.b

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation, which is addressed below in two parts. (EX1002 ¶235.)

(1) providing access to code for use by a client node [], where the code causes the client node to utilize the network application to operate in accordance with the non-TCP protocol to:²⁸

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for at least the same reasons discussed above for limitation 24.b.1. (*Supra* Section IX.A.1.b.1; EX1002 ¶¶236–239.) For example, **remote machine 30** in the *Wookey-Berg* combination sends (“provides access to”) the code (e.g., (1) one of the encoded URLs in the received HTML page 288, or (2) the HTML page 288,

²⁸ See n.7. (EX1002 ¶236 n.26.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

which includes the encoded URLs) to **client machine 10** (“client node”), where the code causes **client machine 10** (e.g., when the client accesses and uses it) to utilize web browser application 280 (“network application”) to operate in accordance with the RUDP protocol (“non-TCP protocol”) to establish an RUDP connection with **remote machine 30’**. (*Supra* Section IX.A.1.b.1; EX1002 ¶¶237–238.)

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 28.b.1. (EX1002 ¶239.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

- (2) [... a client node] including one or more processors in communication with a non-transitory memory storing a network application that is configured to operate in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, []

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for at least the same reasons discussed above for limitation 24.b.2. (*Supra* Section IX.A.1.b.2; EX1002 ¶240.)

- c) **identify idle information for use in detecting an idle time period that results in a non-TCP connection being subject to deactivation;**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶241–251.) As discussed for limitation 24.b, the above-discussed *Wookey-Berg* combination would have included processes that involved establishing an RUDP connection between **client machine 10** and **remote machine 30’**. (*Supra* Section IX.A.1.b.) As discussed above for limitation 24.d and further below, *Berg*’s RUDP protocol includes processes for identifying null segment timeout and transfer state timeout values (individually or collectively, “idle information”) used in detecting an idle time period that results in the RUDP (“non-

Petition for *Inter Partes* Review
Patent No. 10,075,564

TCP”) connection being subject to deactivation. (*Supra* Section IX.A.1.d, EX1002 ¶242.)

For example, *Berg* discloses nodes negotiating timeout values (e.g., null segment timeout value and transfer state timeout value) for the RUDP connection. (*Supra* Section IX.A.1.d; EX1007, 24:40–48.) As explained, the null segment timeout value corresponds to detecting an idle time period that results in the RUDP connection being subject to deactivation. (*Supra* Section IX.A.1.d.1.) As also explained, the transfer state timeout value corresponds to detecting an idle time period that results in the RUDP connection being subject to deactivation. (*Supra* Section IX.A.1.d.2.) Thus, the null segment timeout and transfer state timeout values are a type of “idle information” conveying information regarding connection timeouts. (EX1002 ¶243.)

This understanding of “idle information” is consistent with the ’564 patent. For example, “[i]dle information ... may include ... a duration of time ... [which] may be specified according to various measures of time including seconds.” (EX1001, 11:37–41; EX1002 ¶244.)

Berg explains that when the client initiates a connection, the client’s ULP identifies negotiable parameters for the SYN segment via an application programmer interface (API), where the negotiable parameters indicate the client’s desired

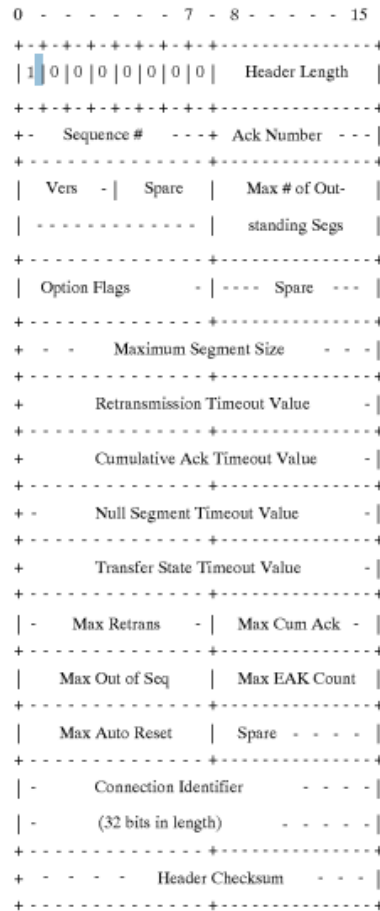
Petition for *Inter Partes* Review
Patent No. 10,075,564

features for the RUDP connection being established. (EX1007, 24:40–48; *see also id.*, 18:57–64.) Therefore, because the client must select and include the desired negotiable parameters in the SYN segment to be sent to the server, the client necessarily identifies these negotiable parameters *prior* to sending the SYN segment. (EX1002 ¶245.)

Berg also discloses that the client identifies these parameters when the server initiates the connection. For example, when the server initiates a reset (EX1007, 23:61–65), the server sends a SYN segment (*id.*, 18:66–67) to negotiate the connection. Here, the client receives the initial SYN segment. The client can then choose to “accept these parameters by echoing them back in its SYN message or propose different parameters in its SYN response.” (*Id.*, 24:43–45.) Thus, the client necessarily identifies each of the parameters upon receiving the SYN segment from the server in order to accept or reject them. (EX1002 ¶¶246–248; *see also supra* Section IX.A.1.d.)

Indeed, “[a]ll configurable parameter *that the peer must know about* are contained in this [SYN] segment” (EX1007, 18:58–61), as shown in Figure 2:

Petition for *Inter Partes* Review
Patent No. 10,075,564



(*Id.*, 19:1–40; EX1002 ¶247.)

To the extent the *Wookey-Berg* combination does not disclose the claimed “identify[ing],” it would have been obvious to modify the functionality in the *Wookey-Berg* system/process such that it identifies the negotiable parameters, as discussed above for limitation 24.d. (EX1002 ¶249; *supra* Section IX.A.1.d.) A POSITA would have been motivated to configure the *Wookey-Berg* system/process to include functionality that identifies the parameters to facilitate the subsequent process of (i) including the client’s desired parameters for the RUDP connection in

Petition for *Inter Partes* Review
Patent No. 10,075,564

the SYN segment or (ii) accepting or rejecting the parameter in the SYN packet consistent with the features described by *Berg*. (EX1007, 24:44–48.) Therefore, to the extent not disclosed, the above-discussed *Wookey-Berg* system/process discloses and/or suggests such features. (EX1002 ¶249.)

Accordingly, for reasons similar to those explained above, a POSITA would have been motivated to implement in the *Wookey-Berg* combined process of establishing the connection between **remote machine 30'** and **client machine 10**, the process of identifying, by **client machine 10**, the null segment timeout value and transfer state timeout value (individually or collectively, “idle information”) for use in detecting an idle time period that results in an RUDP (“non-TCP”) connection being subject to deactivation. (*See supra* Section IX.A.1.b (discussing the *Wookey-Berg* combination).) Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 28.c. (EX1002 ¶¶250–251.)

- d) **generate, based on the idle information, a non-TCP packet including an idle time period parameter field identifying metadata that is specified in a number of seconds or minutes; and**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶252–259.) For example, *Berg*’s RUDP protocol discloses generating, based on the identified null segment timeout and transfer state timeout values (individually or collectively, “idle information”), an RUDP SYN

Petition for *Inter Partes* Review
Patent No. 10,075,564

segment (“non-TCP packet”) including the timeout values, specified in seconds, in an associated parameter field. (EX1002 ¶¶235–257; *supra* Sections IX.A.1.b–d.) This occurs in two scenarios where the client generates a SYN segment: both (i) when the client initiates the connection and (ii) when the server initiates the connection. (EX1002 ¶257.)

As explained for limitation 24.c, *Berg* discloses that the segments exchanged in the RUDP connection are non-TCP packets. (*Supra* Section IX.A.1.c.) The RUDP connection is established by the exchange of SYN segments between the two hosts. (EX1007, 18:57–60, 24:40–47.) The SYN segment includes parameter fields for all of the proposed negotiable parameters for the RUDP connection. (EX1007, 24:40–48; *see also id.*, 18:57–61.) *Berg* thus discloses generating an RUDP SYN segment (“non-TCP packet”) for establishing an RUDP connection. (EX1002 ¶¶254–255.)

As explained for limitation 24.d, the SYN segment includes fields for each of the parameters, including null segment timeout and transfer state timeout values (individually or collectively, “metadata”) that are specified in a number of seconds by the ULP for identifying an idle time period. (*Supra* Section IX.A.1.d; *see also* EX1007, 20:36–42, 20:43–48, 24:40–42, 19:1–40; EX1002 ¶256.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

In the first scenario where the client generates a SYN segment upon initiating the connection, the ULP of the client identifies the negotiable parameters for the SYN segment via an API. (EX1007, 24:40–48; *see also id.*, 18:57–61.) The client then “initiates a connection” by sending the generated “SYN segment which contains the negotiable parameters” (e.g., the identified null segment timeout and transfer state timeout values) to the server. (*Id.*, 24:40–47.) Thus the SYN segment is generated based on the claimed idle information (e.g., the identified null segment timeout and transfer state timeout values). (*Id.*; EX1002 ¶257; *supra* Section IX.A.1.d.)

In the second scenario, the client receives the initial SYN segment from the server. The client can then choose to “accept these parameters by echoing them back in its SYN message or propose different parameters in its SYN response,” i.e., the generated responsive SYN segment is based on the claimed idle information (e.g., the identified null segment timeout and transfer state timeout values). (EX1007, 24:43–45; EX1002 ¶257; *supra* Section IX.A.1.d.)

Thus the client, under either scenario, generates an RUDP SYN segment. (EX1002 ¶257.)

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been further configured to include the process of

Petition for *Inter Partes* Review
Patent No. 10,075,564

generating, by **client machine 10**, based on the null segment timeout value and transfer state timeout value (individually or collectively, “idle information”), an RUDP SYN segment (“non-TCP packet”) including a parameter field (“idle time period parameter field”) identifying the null segment timeout value and transfer state timeout value (individually or collectively, claimed “metadata”) that is specified in a number of seconds or minutes. (*See supra* Section IX.A.1.b.) Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 28.d. (EX1002 ¶¶258–259.)

- e) **send, from the client node to a server node and for setting up the non-TCP connection, the non-TCP packet to provide the metadata to the server node, for use by the server node in determining a timeout attribute associated with the non-TCP connection.**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶260–269.) For example, **client machine 10** sends an RUDP SYN segment (“the non-TCP packet”) to **remote machine 30’** (“server node”) to provide the negotiable parameters, including the null segment timeout and transfer state timeout values (individually or collectively, “metadata”), to establish the RUDP (“non-TCP”) connection, in at least two ways. (*Supra* Section IX.A.5.d.) **Remote machine 30’** uses the received metadata to determine a timeout attribute associated with the RUDP connection, as explained below. (EX1002 ¶¶261–262.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

First, when the client initiates an RUDP connection, the client sends the initial RUDP SYN segment “which contains the negotiable parameters” including the identified null segment timeout and transfer state timeout values (individually or collectively, “metadata”) to the server. (EX1007, 24:40–47.) Second, when the client receives the initial SYN segment from the server, the client can then choose to “accept these parameters by echoing them back in its SYN message or propose different parameters in its SYN response.” (*Id.*, 24:43–45; EX1002 ¶¶263–264.)

In either scenario, the server maintains its own null segment timer (“timeout attribute”) based on a timeout value twice the client’s null segment timeout value (received “metadata” from the SYN message). (EX1007, 23:45–52.) When the server’s null segment timer expires, an “auto reset is initiated.” (EX1007, 23:45–57; EX1002 ¶265; *supra* Section IX.A.1.d.1.) Likewise, in either scenario, the server maintains a transfer state timer (“timeout attribute”) based on the transfer state timeout value (received “metadata”). (EX1002 ¶266; *supra* Section IX.A.1.d.2.) This transfer state timeout value “indicate[s] the amount of time the state information will be saved for a connection after an auto reset occurs” (EX1007, 20:42–45) in the RUDP connection. When the server’s transfer state timer expires, “the connection state will be lost and buffers on the saved queues will be freed.” (EX1007, 24:21–23.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

Thus, the null segment timer and transfer state timer are each a type of a “timeout attribute,” which are configured by the null segment timeout value and transfer state timeout value (provided “metadata”), respectively, associated with the RUDP connection. (EX1002 ¶267.)

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been further configured to include the process of sending, from **machine 10** to **machine 30’** and for setting up the RUDP (“non-TCP”) connection, the RUDP SYN segment (“non-TCP packet”) to provide the null segment timeout value and transfer state timeout value (individually or collectively, “metadata”) to **remote machine 30’**, for use by **remote machine 30’** in determining a null segment timeout timer and transfer state timeout timer (individually or collectively, “timeout attribute”) associated with the RUDP connection. (*See supra* Section IX.A.1.b.) Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 28.e. (EX1002 ¶¶268–269.)

6. Claim 29

- a) **“The computer-implemented method of claim 28, and further comprising: providing access to additional code that causes the client node to utilize the network application to communicate with another server node in accordance with a TCP protocol using a hypertext transfer protocol (HTTP) and further using a TCP connection that is different than the non-TCP connection.”**²⁹

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for reasons similar to those given above for limitation 24.f. (EX1002 ¶¶270–275; *see supra* Sections IX.A.1.f, IX.A.2–3.) For example, *Wookey* discloses that **remote machine 30** (“another server node”) causes code (e.g., (1) one of the encoded URLs in the HTML page 288, or (2) the HTML page which includes the encoded URLs) to be sent to **client machine 10** (“client node”) over a TCP/IP connection. (*Supra* Sections IX.A.1.b, IX.A.1.f.) This code causes **client machine 10** via the HTTP-based web browser (“network application”) to access various resources by communicating with remote machines when **client machine 10** uses the code. (*Supra* Sections IX.A.1.b, IX.A.1.f; EX1002 ¶272.)

Regarding “providing access to additional code,” one of the encoded URLs in the provided HTML page 288 can point to **remote machine 30** itself (i.e., the desired

²⁹ *See* n.20. (EX1002 ¶270 n.27.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

resource is available on **remote machine 30**). (*Supra* Sections IX.A.1.f; *see also supra* Section IX.A.1.b.) Thus, as explained for limitation 24.f, this other encoded URL identifying a resource located on **remote machine 30** (“another server node”) is “additional code.” (*Supra* Section IX.A.1.f.) When **client machine 10** selects this resource, a new connection is established with **remote machine 30**. (*Id.*; EX1002 ¶273; *supra* Section IX.A.1.f.) This new connection between **client machine 10** and **remote machine 30** may be made over a TCP/IP connection using HTTP. (*Supra* Sections IX.A.1.b.2, IX.A.1.f, IX.A.3.) This TCP connection between **remote machine 30** and **client machine 10** is different from the RUDP connection between **remote machine 30**’ and **client machine 10** for similar reasons as discussed above for limitations 24.b and 24.f. (EX1002 ¶274; *supra* Sections IX.A.1.b, IX.A.1.f.)

Thus, the *Wookey-Berg* combination discloses and/or suggests claim 29. (EX1002 ¶275.)

X. DISCRETIONARY DENIAL IS NOT APPROPRIATE

The Board should not exercise its discretion to deny institution under 35 U.S.C. § 314(a). *NHK Spring Co., Ltd. v. Intri-Plex Techs, Inc.*, IPR2018-00752, Paper 8 (Sept. 12, 2018) does not apply here as no trial date has been scheduled in the related litigation. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 3 (Mar. 20, 2020) (precedential) (“*NHK* applies ... where the district court has set a trial date to occur earlier than the Board’s deadline to issue a final written decision.”). The six-factor test addressed in *Fintiv* (“*Fintiv* factor(s)”) favors institution. *See id.*, 5–6.

The **first *Fintiv* factor** (stay) is neutral, if not weighing in favor of institution. While there is currently no stay pending IPR resolution before the Western District of Texas (“WDTX”) court, Google intends to pursue such a stay should institution be granted. The Board should not speculate as to how the court would rule on the requested stay based on actions taken in different cases. *Western Digital Corp. et al. v. Martin Kuster*, IPR2020-01391, Paper 10 at 8-9 (February 16, 2020) (finding this factor neutral despite Patent Owner assertions regarding WDTX Judge Albright’s past practices regarding stays). Moreover, Google filed a motion to transfer in the district court litigation. The Court has currently stayed the litigation pending resolution of Google’s transfer motion. (EX1030.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

The **second *Fintiv* factor** (proximity of trial dates) weighs strongly in favor of institution. *First*, the district court has not set a trial date, which “weighs significantly against exercising [] discretion to deny institution of the proceeding.” *Google LLC v. Uniloc 2017 LLC*, IPR2020-00441, Paper 13 at 35 (July 17, 2020); *Nested Bean, Inc. v. Big Beings USA PTY LTD*, IPR2020-01234, Paper 15 at 14 (Jan. 25, 2021) (same). In fact, while the parties submitted proposed schedules, the court granted Google’s motion to stay the case pending resolution of Google’s transfer motion and declined to set a case schedule.³⁰ (EX1030; EX1024.) Accordingly, determining at this time whether the Board’s anticipated FWD date (e.g., Sept.-Oct. 2022) will occur after trial in the related litigation would be speculative.

Even if the parties’ most recent proposed schedule was considered as a baseline for comparison (which Google submits would be improper at this time), dates in that proposed schedule would need to be delayed because they were proposed prior to the court’s recent stay order and did not take the stay into consideration. (EX1030; EX1024, 5-9.) Accordingly, the proposed dates in light of

³⁰ Disposition of Google’s transfer motion has taken priority over other activities. (EX1030.) *See also In re Apple Inc.*, 979 F.3d 1332, 1337 (Fed. Cir. 2020) (“disposing of a filed transfer motion ‘should unquestionably take top priority.’”).

Petition for *Inter Partes* Review
Patent No. 10,075,564

the court’s stay order demonstrates that trial will likely occur after August 2022.³¹ (EX1024, 6, 9.) Thus, any projected trial date (*albeit* speculative) would be more likely after the expected due date of the Board’s FWD (e.g., around September-October 2022).

The Board has instituted trial even where an *actual* litigation trial date precedes the FWD deadline by two months or more. *Sand Revolution II, LLC v. Cont’l Intermodal Grp–Trucking LLC*, IPR2019-01393, Paper 24 at 8-10 (June 16, 2020) (“*Sand*”) (informative) (factor 2 favored institution despite WDTX trial preceding FWD deadline by five months); *Western Digital Corp.*, IPR2020-01391 at 9-10 (factor 2 found to be neutral despite WDTX trial predating FWD by three and a half months); *Apple Inc. v. Parus Holdings, Inc.*, IPR2020-00686, Paper 9 at 11–13, 22 (instituting trial where WDTX trial was two months before FWD deadline); *SMIC, Am. v. Innovation Foundry Techs., LLC*, IPR2020-00786, Paper

³¹ Consistent with the court’s practice, the proposed order stages deadlines and trial date based on the proposed August 2021 *Markman* hearing and that the “actual trial date” may materially differ from the schedule and that the court would consider “reasonable amendments to the case schedule post-*Markman*.” (EX1029, 9 n.3; *id.*, 6.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

10, at 20–21 (Oct. 5, 2020) (same, where WDTX trial date preceded FWD by one month).

It is also unlikely that trial in WDTX will proceed without delay. WDTX has issued a suspension order every month for the past eleven months suspending almost all trials in the district from March 13, 2020 to at least March 31, 2021 due to the COVID-19 pandemic, creating a large backlog of trials. (EX1025; *see also* EX1026, 2 (agreeing to stipulation to postpone trial date “to allow the COVID-19 situation to ameliorate”); EX1029 (showing J. Albright’s docket including 792 cases filed in 2020).) Thus, as the Board has recognized, civil trials in WDTX “may possibly slip ... [due to] months of backlogged trials, including many active criminal cases that would take precedence over civil trials.” *HP Inc. v. Slingshot Printing LLC*, IPR2020-01085, Paper 12 at 7 (Jan. 14, 2021); *Sand* at 8-10.

Finally, the potential for transfer based on Google’s transfer motion further reduces the likelihood that trial will precede the Board’s FWD. *Dish Network, L.L.C. v. Broadband iTV, Inc.*, IPR2020-01267, Paper 15 at 17-18 (Jan. 21, 2021) (explaining that the second *Fintiv* factor “is a proxy for the *likelihood* that the trial court will reach a decision on validity issues before the Board reaches a [FWD]”). Indeed, transfer of the WDTX case to the proper venue (the Northern District of California) will undoubtedly delay any trial beyond the anticipated FWD due date.

Petition for *Inter Partes* Review
Patent No. 10,075,564

C.f. Google LLC v. Uniloc 2017 LLC, IPR2020-00115, Paper 8, at 7 n.7 (Mar. 27, 2020) (denying institution based in part on scheduled trial date that was later vacated upon transfer); *see Uniloc 2017 LLC v. Google LLC*, Case No. 18-CV-00502, Dkt No. 277 (E.D. Tex. June 19, 2020) (transfer order vacating unreach deadlines).

The **third *Fintiv* factor** (investment in parallel proceedings) also weighs in favor of institution. The case is in earliest stages, and as discussed, all deadlines other than those concerning Google’s transfer motion have been stayed. Thus, investment by the parties in invalidity and claim constructions issues will be delayed, especially if the case is transferred. And even if the transfer motion is denied, using the above-discussed proposed schedule as an exemplary baseline (which will require adjustments), any *Markman* hearing in the case is likely not to occur until August 2021 or later, and thus would occur near the expected due date of the Board’s institution decision. (EX1024, 6.) Discovery will not commence, and dispositive motions will not be due, until after the *Markman* hearing. Other case activity that has taken place or will eventually take place (outside claim construction) that does not relate to validity should “not weigh in [the Board’s] consideration of this issue.” *Western Digital Corp.*, IPR2020-01391 at 11. Such minimal investment is insufficient to support discretionary denial. *Juniper Networks, Inc. v. Huawei Digital Techs. (Cheng Du) Co., Ltd.*, IPR2020-01130, Paper 13 at 12–13 (Jan. 22,

Petition for *Inter Partes* Review
Patent No. 10,075,564

2021) (factor 3 weighs against discretionary denial when claim construction order has been entered, but fact and expert discovery has not closed, and dispositive motions and briefings have not been submitted); *Dish* at 19-21 (similar); *HP* at 7 (similar). Additionally, Google’s diligence in filing this Petition just three months after receiving PO’s narrowed list of asserted claims³² further weighs against discretionary denial. *Dish* at 20–21 (petitioner’s diligence in filing the petition weighed against discretionary denial); *Fintiv* at 11.

The **fourth *Fintiv* factor** (overlap) is neutral if not in favor of institution. There is, at this early stage of the litigation, no evidence of overlap with the grounds here and any invalidity positions to be pursued in the district court. Indeed, preliminary invalidity contentions have not yet been served. And, as discussed

³² PO initially asserted 455 claims across eight patents (including 29 claims from the ’564 patent). (EX1017, 1.) The WDTX court ordered PO to substantially reduce the asserted claims to approximately 65 claims. On October 20, 2020, PO narrowed its asserted claims as ordered, but “reserve[d] the right to either narrow the claims further, **substitute claims**, or **add a reasonable number of claims**.” (EX1027.) On December 4, 2020, PO informed Petitioner that it will not modify its narrowed list of asserted claims “**at this time**.” (EX1028.)

Petition for *Inter Partes* Review
Patent No. 10,075,564

above, the district court must first resolve the pending transfer motion before addressing substantive issues, including invalidity. Given the stay of the litigation, it is likely expert discovery relating to invalidity will not be completed until at the earliest mid-2022 and dispositive motions on those issues would be due after that. Accordingly, substantive invalidity issues would not be addressed in the litigation until well past the institution decision and likely very near the time of the Board's expected FWD. Moreover, Petitioner may rely on invalidity grounds not asserted here in the litigation, including system art and prior art references and combinations not raised herein. Therefore, any potential overlap between this proceeding and the district court is minimal and speculative at this time.

Other circumstances (*Fintiv* factor six) also favor institution. The Petition presents strong grounds demonstrating the unpatentability of the challenged claims, which weighs against discretionary denial. (*See supra* Section IX.) *Western Digital Corp.*, IPR2020-01391 at 14-15. Indeed, the Board has already instituted the Google -845 IPR and the Unified -742 IPR challenging claims of the related '995 patent. (*See supra* Section II.) The Unified -742 IPR is based in part on *Berg*, which is being applied here. Moreover, this petition and the concurrently-filed sister petition (*id.*) are the only challenges to the '564 patent before the Board, which is a "crucial

Petition for *Inter Partes* Review
Patent No. 10,075,564

fact” favoring institution. *Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 10 at 6 (May 12, 2020).³³

While ***Fintiv* factor 5** (same parties) may weigh slightly in favor of denial, the remaining factors are at least neutral, if not strongly favoring institution. Further, even if the Board finds that **factor 1** (or even **factor 2**) somehow tilts in favor of denial, the investment of resources, lack of evidence of overlap, and/or strength of Petitioner’s unpatentability positions (**factors 3, 4, 6**) outweigh these other factors. Accordingly, based on a “holistic view” of whether the system’s integrity and efficiency is best served, institution is proper. *Samsung Elecs. Co. Ltd. v. Dynamics Inc.*, IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

³³ Discretionary denial under *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper No. 19 is not applicable here given the ’564 patent is not at issue in any other proceeding pending before the Board (outside of Petitioner’s concurrently-filed sister petition,).

Petition for *Inter Partes* Review
Patent No. 10,075,564

XI. CONCLUSION

For the reasons provided, Petitioner requests institution of IPR for claims 24–29 of the '564 patent based on the ground specified in this petition.

Respectfully submitted,

Dated: March 15, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,075,564

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,075,564 contains, as measured by the word-processing system used to prepare this paper, 13,989 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: March 15, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,075,564

CERTIFICATE OF SERVICE

I hereby certify that on March 15, 2021, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,075,564 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on PAIR:

Patrick Caldwell (Reg. No. 44580)
G. Gordon (Reg. No. 64517)
P.O. Box 59655
Dept. SVIPGP
Dallas, TX 75229

A courtesy copy was also sent via electronic mail to the Patent Owner's litigation counsel at the following addresses:

TD-PTAB@devlinlawfirm.com
ddahlgren@devlinlawfirm.com
dlflitparas@devlinlawfirm.com

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)

EXHIBIT 5

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

JENAM TECH, LLC,
Patent Owner

Patent No. 10,075,565

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,075,565**

Petition for *Inter Partes* Review
Patent No. 10,075,565

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	1
III.	PAYMENT OF FEES	2
IV.	GROUND FOR STANDING.....	2
V.	PRECISE RELIEF REQUESTED	3
	A. Claims for Which Review Is Requested	3
	B. Statutory Ground of Challenge	3
VI.	LEVEL OF ORDINARY SKILL	4
VII.	OVERVIEW OF THE '565 PATENT	4
VIII.	CLAIM CONSTRUCTION	8
IX.	DETAILED EXPLANATION OF GROUNDS.....	10
	A. Ground 1: The Combination of <i>Wookey</i> and <i>Berg</i> Render Obvious Claims 25 and 28	10
	1. Claim 25	10
	2. Claim 28.....	62
X.	DISCRETIONARY DENIAL IS NOT APPROPRIATE	71
XI.	CONCLUSION.....	79

Petition for *Inter Partes* Review
Patent No. 10,075,565

LIST OF EXHIBITS

EX1001	U.S. Patent No. 10,075,565
EX1002	Declaration of Bill Lin, Ph.D.
EX1003	Curriculum Vitae of Bill Lin, Ph.D.
EX1004	Prosecution History of U.S. Patent No. 10,075,565
EX1005	U.S. Pre-Grant Publication No. 2007/0171921 to Wookey <i>et al.</i>
EX1006	RESERVED
EX1007	U.S. Patent No. 6,674,713 to Berg <i>et al.</i>
EX1008	DARPA RFC 793 TRANSMISSION CONTROL PROTOCOL
EX1009	U.S. Patent No. 6,584,546 to Kavipurapu
EX1010	U.S. Patent No. 9,923,995
EX1011	Bova <i>et al.</i> , RELIABLE UDP PROTOCOL <draft-ietf-sigtran-reliable-udp-00.txt> 25 February 1999
EX1012	U.S. Pre-Grant Publication No. 2004/0093376 to De Boor <i>et al.</i>
EX1013	U.S. Patent No. 7,535,913 to Minami <i>et al.</i>
EX1014	U.S. Pre-Grant Publication No. 2005/0204013 to Raghunath <i>et al.</i>
EX1015	U.S. Pre-Grant Publication No. 2007/0005804 to Rideout
EX1016	U.S. Pre-Grant Publication No. 2004/0098748 to Bo <i>et al.</i>
EX1017	<i>Jenam Tech, LLC's First set of Infringement Contentions</i> regarding U.S. Patent No. 10,075,565 (August 21, 2020)
EX10181	IETF RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1
EX1019	U.S. Patent No. 7,636,805 to Rosenberg
EX1020	U.S. Patent No. 6,212,175 to Harsch

Petition for *Inter Partes* Review
Patent No. 10,075,565

EX1021	U.S. Patent No. 8,259,716 to Diab
EX1022	U.S. Patent No. 6,665,727 to Hayden
EX1023	U.S. Patent No. 6,981,048 to Abdolbaghian <i>et al.</i>
EX1024	<i>Jenam Tech., LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Joint Submission of Proposed Amended Scheduling Order, ECF No. 57 (W.D. Tex. Mar. 4, 2021)
EX1025	Thirteenth Supplemental Order Regarding Court Operations Under the Exigent Circumstances Created By the COVID-19 Pandemic, The United States District Court for the Western District of Texas (February 2, 2021)
EX1026	<i>Digital Retail Apps, Inc. v. H-E-B, LP</i> , Case No. 6:19-cv-00167, Joint Stipulation and Order Postponing Trial, ECF No. 182 (W.D. Tex. Jan. 13, 2021)
EX1027	<i>Jenam Tech, LLC's</i> Preliminary Narrowing of Claims (October 20, 2020)
EX1028	<i>Jenam Tech, LLC's</i> Correspondence Regarding Narrowing of Claims (December 4, 2020)
EX1029	Judge Alan D. Albright's Case Statistics By Year (Retrieved from DocketNavigator on March 9, 2021)
EX1030	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Order Granting Motion to Stay Case, ECF No. 58 (W.D. Tex. Mar. 10, 2021)

Petition for *Inter Partes* Review
Patent No. 10,075,565

I. INTRODUCTION

Google LLC (“Petitioner”) requests *inter partes* review (“IPR”) of claims 25 and 28 (“the challenged claims”) of U.S. Patent No. 10,075,565 (“the ’565 patent”) (EX1001) assigned to Jenam Tech, LLC (“Patent Owner” or “PO”). For the reasons below, the challenged claims should be found unpatentable and canceled.

II. MANDATORY NOTICES

Real Parties-in-Interest: Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies the following as the real parties-in-interest: Google LLC.¹

Related Matters: The ’565 patent is asserted in the following civil actions: *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.); and *Jenam Tech, LLC v. Samsung Group*, Case No. 4:20-cv-00279 (E.D. Tex.) (voluntarily dismissed).

The ’565 patent claims priority to U.S. Patent Application No. 12/714,454 filed February 27, 2010. (EX1001, Cover.) U.S. Patent No. 9,923,995 (“the ’995 patent”) also claims priority to U.S. Patent Application No. 12/714,454. (EX1010,

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

Petition for *Inter Partes* Review
Patent No. 10,075,565

Cover.) The '995 patent is subject to the following instituted IPRs: *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845 (“Google -845 IPR”); and *Unified Patents, LLC v. Jenam Tech, LLC*, IPR2020-00742 (“Unified -742 IPR”).

Petitioner is also concurrently filing IPR petitions challenging U.S. Patent Nos. 10,075,564 and 10,375,215, which are also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.).

Counsel and Service Information: Lead counsel: Naveen Modi (Reg. No. 46,224). Backup counsel: (1) Joseph E. Palys (Reg. No. 46,508), (2) Quadeer A. Ahmed (Reg. No. 60,835), (3) Jason Heidemann (Reg. No. 77,880).

Service information is Paul Hastings LLP, 2050 M Street NW, Washington, DC 20036, Tel.: 202.551.1700, Fax: 202.551.1705, email: PH-Google-Jenam-IPR@paulhastings.com. Petitioner consents to electronic service.

III. PAYMENT OF FEES

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-2613.

IV. GROUNDS FOR STANDING

Petitioner certifies that the '565 patent is available for review, and Petitioner is not barred/estopped from requesting review on the grounds herein.

Petition for *Inter Partes* Review
Patent No. 10,075,565

V. PRECISE RELIEF REQUESTED

A. Claims for Which Review Is Requested

Petitioner requests review and cancellation of claims 25 and 28 as unpatentable based on the following ground.

B. Statutory Ground of Challenge

Ground 1: Claims 25 and 28 are rendered obvious under 35 U.S.C. § 103(a) over U.S. Publication No. 2007/0171921 to Wookey *et al.* (“*Wookey*”) (EX1005) in view of U.S. Patent No. 6,674,713 to Berg *et al.* (“*Berg*”) (EX1007).²

The ’565 patent issued from Application No. 15/915,052, filed on March 7, 2018, which claims priority through a number of applications back to Application No. 12/714,454, filed on February 27, 2010. (EX1001, 1:8–28.) Petitioner assumes for this proceeding only, without conceding, that the earliest effective filing date of the ’565 patent is February 27, 2010.

Wookey was published on July 26, 2007, from an application filed on November 14, 2006. (EX1005, Cover.) *Berg* was issued on January 6, 2004, from an application filed on February 23, 1999. (EX1007, Cover.) Thus, *Wookey* and

² Other references identified herein are provided to show the state of the art at the time of the alleged invention of the ’565 patent.

Petition for *Inter Partes* Review
Patent No. 10,075,565

Berg qualify as prior art at least under 35 U.S.C. § 102(b) (pre-AIA). Neither reference was considered during prosecution. (EX1001, Cover; *generally* EX1004).

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art at the time of the alleged invention of the '565 patent ("POSITA") would have had an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking. (EX1002 ¶¶16–18.)³ More education can supplement practical experience and vice versa. (*Id.*)

VII. OVERVIEW OF THE '565 PATENT

The '565 patent is directed to networking, and to the sharing of information for detecting an idle TCP connection. (EX1001, 2:27–29, 8:6–27; EX1002 ¶¶64–69.) Figure 7 illustrates such a process.

³ Petitioner submits the declaration of Dr. Bill Lin (EX1002), an expert in the field of the '565 patent. (EX1002 ¶¶3–11; *see also id.* ¶¶12–15; EX1003.)

Petition for *Inter Partes* Review
 Patent No. 10,075,565

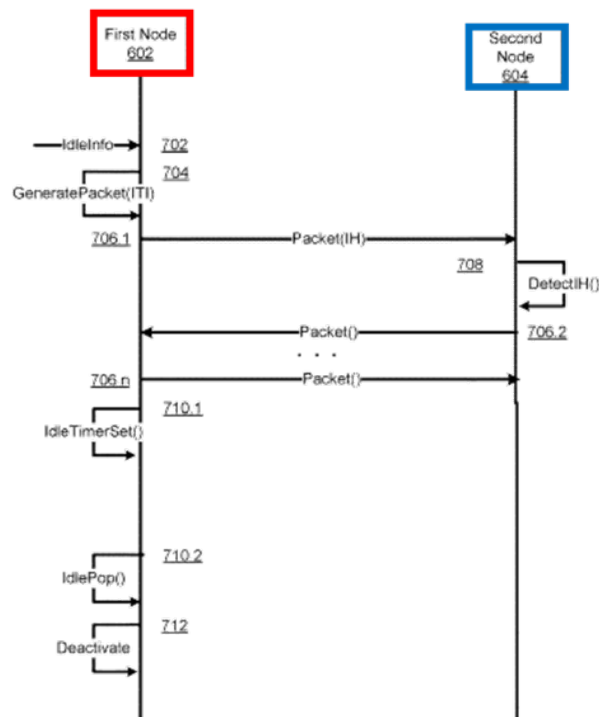


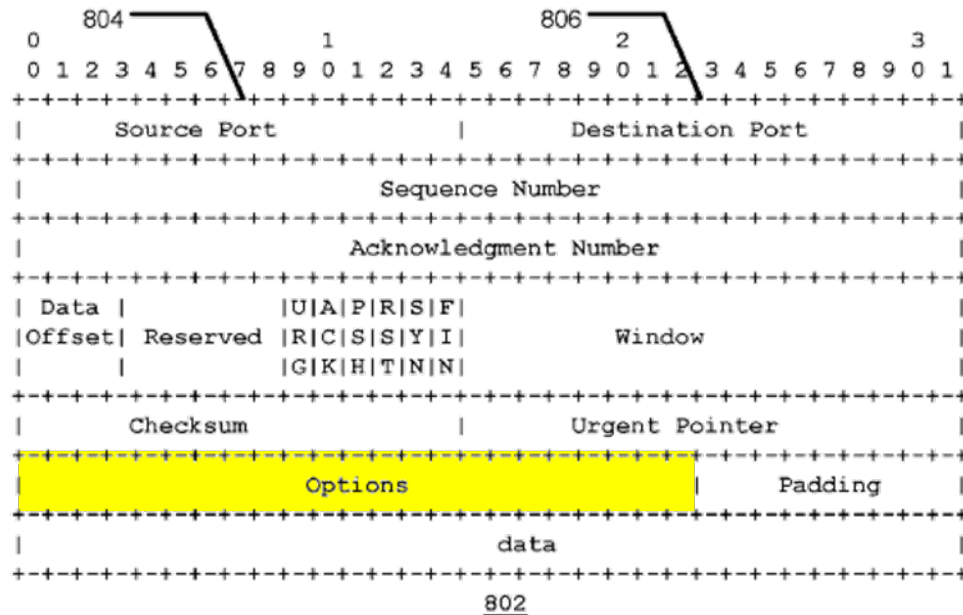
Fig. 7

(*Id.*, FIG. 7 (annotated); EX1002 ¶64.)

First, the **first node 602** receives a message 702 that identifies idle information representing a duration for an idle time period (“ITP”). (EX1001, 11:30–39.) Message 702 may take various forms, such as “a message received via a network.” (*Id.*, 11:34–39.) The idle information “may include and/or identify a duration of time” for detecting an ITP. (*Id.*, 12:12–18.) The duration “may be specified according to various measures of time[,], including seconds, minutes, hours, and/or days.” (*Id.*)

Petition for *Inter Partes* Review
Patent No. 10,075,565

Next, the **first node** “generat[es] a TCP packet including an ITP header based on received idle information.” (*Id.*, 15:45–48.) The **TCP options field** of a TCP packet may store the ITP header. (*Id.*, FIG. 8) Figure 8 below, “adapted from RFC 793,” illustrates this configuration. (*Id.*, 6:5–8, 15:20–28.)



Petition for *Inter Partes* Review
 Patent No. 10,075,565

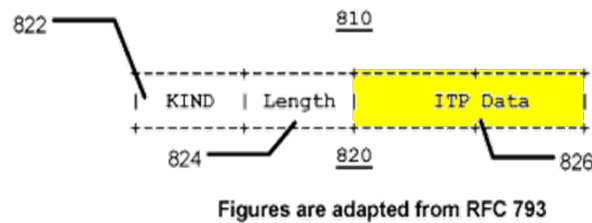


Fig. 8

(*Id.* (cropped-annotated); EX1002 ¶¶67.) The **TCP options field** is used to carry ITP information. (EX1001, 15:22–28.) The ITP header can also be “in structures and locations [in a TCP packet] other than those specified for TCP options in RFC 793” (*Id.*, 14:55–58).

The ITP header is exchanged during the three-way handshake (EX1001, 14:25–39.) For example, the **first node** transmits a message 706.1, i.e., a TCP packet including an ITP header (IH) containing **ITP information**, to the **second node 604**. (*Id.*, 16:3–5; FIG. 7.) Message 708 exemplifies the **second node**’s detection of the ITP header in the received TCP packet. (*Id.*, 21:25–29; EX1002 ¶¶67–69.)

All the limitations in the challenged claims were known in the prior art and obvious. (*See* Section IX; *see also* EX1002 ¶¶69, 19–63 (technology background), citing EXS. 1008–1009, 1011–1016, 1018–1023.)

VIII. CLAIM CONSTRUCTION

Under the applicable standard in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), claim terms are typically given their ordinary and customary meanings as understood by a POSITA at the time of the invention based on the claim language, specification, and the prosecution history of record. *Phillips*, 415 F.3d at 1313; *see also id.* at 1312–16. The Board, however, only construes the claims when necessary to resolve the controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper No. 11 at 16 (Aug. 14, 2015) (citation omitted). Petitioner believes no express constructions of any claim terms are necessary to assess whether the prior art reads on the challenged claims. (EX1002 ¶70.)

Claims 25 and 28 recite the term “[a/the] second protocol that is different from the TCP.” (EX1001, 26:56–57, 28:1–2.) The ’565 patent mentions “different from the TCP” only in the “Summary” section, which the ’565 patent explains is “not an extensive overview of the disclosure and it does not identify key/critical elements of the invention or delineate the scope of the invention.” (*Id.*, 2:33–40.) As discussed below, *Berg*’s RUDP protocol is different from the TCP protocol. (*See, e.g., infra* Section IX.A.1.e.)

Moreover, the Board preliminarily found *Berg* discloses a non-TCP protocol in the Unified -742 IPR. *See, e.g.,* Unified -742 IPR, Paper 11 (Institution Decision)

Petition for *Inter Partes* Review
Patent No. 10,075,565

at 10–11. Thus, because Petitioner relies on *Berg*’s similar disclosures to satisfy the “protocol that is different from the TCP” terms herein (*see infra* Sections IX.A.1.e, IX.A.2.b.), and given the ’565 patent offers no evidence requiring a special meaning of this term and the prior art discloses this feature under any reasonable interpretation, construction of this term is unnecessary.⁴

⁴ Petitioner reserves all rights to raise claim construction and other arguments, including challenges under 35 U.S.C. §§ 101 or 112, in district court as relevant to those proceedings. *See, e.g., Target Corp. v. Proxicom Wireless, LLC*, IPR2020-00904, Paper 11 at 11–13 (November 10, 2020). A comparison of the claims to any accused products in litigation may raise controversies that are not presented here given the similarities between the references and the patent.

IX. DETAILED EXPLANATION OF GROUNDS**A. Ground 1: The Combination of *Wookey* and *Berg* Render Obvious Claims 25 and 28****1. Claim 25****a) An apparatus comprising:**

To the extent limiting, *Wookey* discloses the limitations of this preamble. (EX1002 ¶¶95–102; *see also id.* ¶¶71–80 (*Wookey* overview).)

Figure 1 below shows an exemplary environment in which client machines (e.g., **client machine 10**) access computer resources provided by a server farm 38 (“apparatus”) over a network connection 150. (EX1005 ¶[0136]; *see also id.* ¶¶[0002], [0016], [0020], [0174], Abstract, FIG. 1; EX1002 ¶97.) The server farm 38 includes multiple remote machines, e.g., **remote machine 30** and **remote machine 30’** (EX1005 ¶[0136]), which are “typical computers” (*id.* ¶[0021], FIGS. 1A–1B). Figures 2C and 3D below additionally show an exemplary environment in which remote machines **30** and **30’** are employed to permit a **client machine 10** access to a computer resource. (*Id.* ¶¶[0016], [0020], [0196].)

Petition for *Inter Partes* Review
Patent No. 10,075,565

FIG. 1

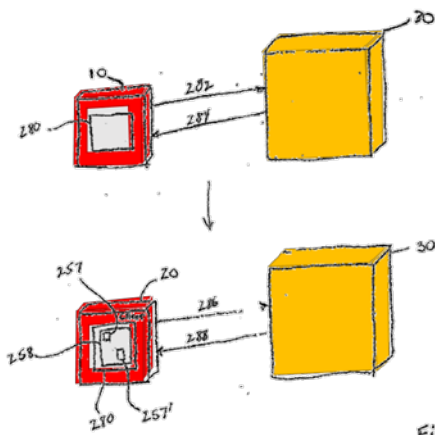
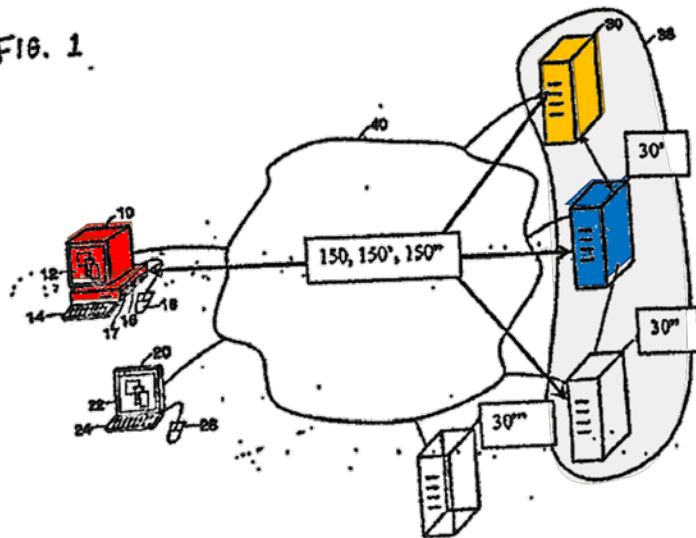


Fig. 2C

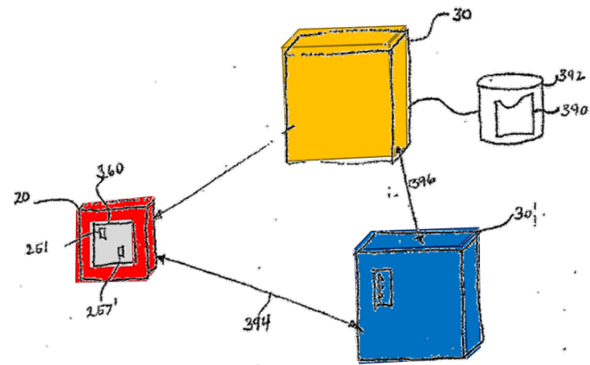


Fig. 3D

(*Id.*, FIGS. 1, 2C, 3D (each annotated) EX1002 ¶¶97–100.)⁵

⁵ *Wookey's* disclosures (including its figures) include certain typographical errors when referencing its client machines (referring to them interchangeably as 10, 10'

Petition for *Inter Partes* Review
Patent No. 10,075,565

The server farm is an apparatus as claimed because, as explained below, it includes a server computer having the features like those recited in claim 25. (EX1002 ¶102; *infra* Sections IX.A.1.b–g.)

b) a server computer including: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, []

Wookey discloses this limitation. (EX1002 ¶¶103–108.) For instance, *Wookey*’s server farm (“apparatus”) includes a group of remote machines (e.g., **remote machine 30** and **remote machine 30’**.) (*Supra* Section IX.A.1.a.) *Wookey* explains that **remote machine 30** is a server computer configured with typical computer architecture, as shown in Figures 1A and 1B below (EX1005 ¶¶[0016], [0021], [0136], [0150], [0160] (“the remote machines 30 ... are provided as

or 20). (EX1002 ¶101; EX1005 ¶¶[0135]–[0136], [0152]–[0154], [0192]–[0196], [0207]–[0216], FIGS. 1, 2C, 3D.) For example, the bottom figure in Fig. 2C erroneously shows client machine “20,” but the corresponding disclosure for both figures in Fig. 2C describes client machine “10.” (EX1005 ¶¶[0192]–[0196].) Moreover, in context of *Wookey*’s disclosures, a POSITA would have understood that any teachings regarding one client machine 10/10’/20 in *Wookey* apply equally to the other client machines. (EX1002 ¶101.)

Petition for *Inter Partes* Review
 Patent No. 10,075,565

computers or computer servers”); EX1002 ¶104.) In this architecture, **remote machine 30** includes a main processor 102 (“one or more processors”) in communication (e.g., via a system bus 120) with a main memory unit 104 and a cache memory 140 (individually or collectively, the claimed “non-transitory memory”). (EX1005 ¶¶[0161]–[0165].)

FIG. 1A

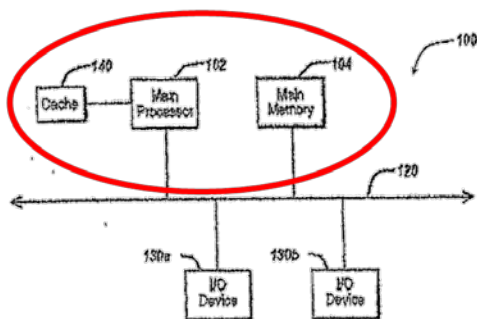
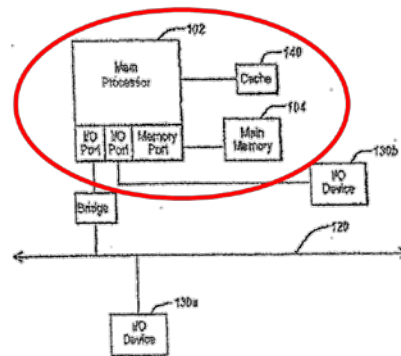


FIG. 1B



(*Id.*, FIGS. 1A and 1B (annotated); EX1002 ¶¶104–105.)

Main memory 104 and cache memory 140 store instructions. For instance, “instructions [are] fetched from the main memory unit 104” by the main processor 102.⁶ (EX1005 ¶[0162]; *id.* ¶[0158], FIGS. 1A–1B.)⁷ Main memory 104 can

⁶ Emphasis added unless otherwise noted.

⁷ *Wookey* interchangeably uses “main processor 102,” “microprocessor 102,” “processor 102,” and “central processing unit 102”. (EX1002 ¶105 n.4; EX1005

Petition for *Inter Partes* Review
Patent No. 10,075,565

include memory chip(s) (e.g., SRAM) that store data accessed by processor 102. (*Id.* ¶[0163].) Also, processor 102 can communicate with cache memory 140. (*Id.* ¶[0165].) As was known in the art, cache memory often stores data and instructions accessible to a processor. (EX1002 ¶¶105–106 (*citing* EX1009, 1:15–39).) Thus, a POSITA would have understood *Wookey*’s cache 140 necessarily stores instructions accessible to processor 102. (EX1002 ¶106.)

Thus **remote machine 30** (“server computer”) includes a “non-transitory memory” (e.g., cache memory 140 and/or main memory unit 104) storing instructions, and is in communication with processor 102 (“one or more processors”), which as explained below, executes the instructions to perform processes consistent with the claimed features. (*Id.* ¶¶107–108; *infra* Sections IX.A.1.c–h.)

c) [] wherein the one or more processors execute the instructions such that a network application operates in accordance with a first protocol including a transmission control protocol (TCP), []

Wookey discloses this limitation. (EX1002 ¶¶109–120.) As explained, processor 102 (“processor”) of **remote machine 30** (“server computer”) fetches

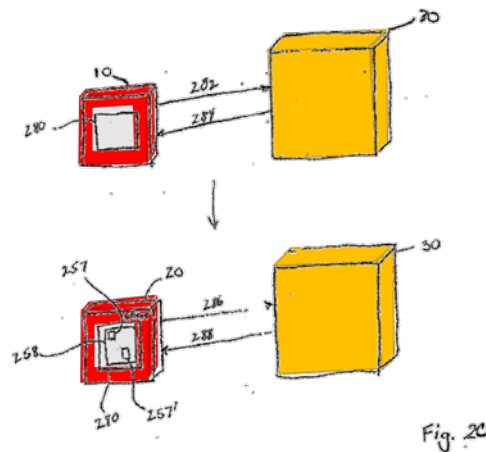
¶¶[0161] (“a central processing unit 102”), [0163] (“microprocessor 102”), [0164] (“processor 102”), FIGS. 1A–B.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

instructions from a “non-transitory memory” (e.g., cache memory 140 and/or main memory unit 104). (*Supra* Section IX.A.1.b.) Main processor 102 executes these instructions to enable **remote machine 30** to act as a web server (“network application”) which operates in accordance with a first protocol including a transmission control protocol (TCP). (EX1002 ¶109.)

For example, **remote machine 30** will manage client machines’ requests through a TCP connection when acting as a web server. (*Id.* ¶¶110–111.) A web server is an example of a network application because it includes software that runs on one computer (e.g., **remote machine 30**) which provides communication to another application (e.g., web browser 280) running on another computer (e.g., **client machine 10**). (*Id.*; EX1005 ¶¶[0179], [0207], [0213].) With respect to Figure 2C, *Wookey* describes an arrangement where **remote machine 30** “acts as a web server” at least because it communicates with **client machine 10** (e.g., via communication link 150) to authenticate, and provide a list of resources to, **client machine 10**. (EX1005 ¶[0207], [0213].)

Petition for *Inter Partes* Review
 Patent No. 10,075,565



(*Id.*, FIG. 2C (annotated); EX1002 ¶111.)

Client machine 10 executes a web browser application 280 (EX1005 ¶[0192]) that “transmits a request 282 to access a Uniform Resource Locator (URL) address corresponding to an HTML page residing on” **remote machine 30** (*id.* ¶[0193]). (*Id.* ¶¶[0024], [0207], [0213].) The request is received and handled by **remote machine 30**. (*E.g.*, *id.* ¶¶[0193]–[0196].) For example, **remote machine 30** includes functionality for authenticating (*id.* ¶[0194]) and preparing-transmitting an HTML page to **client machine 10** (*id.* ¶[0196]). (*See also id.* ¶¶[0179], [0207], [0213]–[0215], [0803]–[0808] (web server functionality), FIGS. 2C, 3D; EX1002 ¶¶112–114.)

The communication link 150 between **client machine 10** and **remote machine 30** uses the TCP. (EX1005 ¶¶[0154]–[0156], [0174], [0731]–[0732].)

Petition for *Inter Partes* Review
Patent No. 10,075,565

Thus, in the communication between **client machine 10** and **remote machine 30** for Figure 2C, **remote machine 30** acts as a web server that operates in accordance with a first protocol including TCP, as claimed. (EX1002 ¶115.)

Thus, *Wookey*'s **remote machine 30**, which can perform processes over a TCP connection to handle HTML-based requests from **client machine 10**, executes instructions via main processor 102 such that its web server ("network application") operates in accordance with a first protocol including a TCP. (*Id.* ¶¶116–118.) Given processor 102 performs the processing functionalities of **remote machine 30** (*supra* Section IX.A.1.b; EX1005 ¶[0021], FIGS. 1A–1B), the main processor 102 would thus execute the instructions stored in the above-described memory such that **remote machine 30** acts as a web server ("network application") in accordance with a first protocol including a TCP, to process, e.g., the received request 282 from **client machine 10**. (EX1002 ¶¶119–120.)

Petition for *Inter Partes* Review
 Patent No. 10,075,565

d) [] the server computer, when operating in accordance with the first protocol to set up a TCP connection, configured to:

communicate a segment including at least one first synchronize bit;
 communicate a first acknowledgement of the segment,
 and at least one second synchronize bit; and
 communicate a second acknowledgement;

Wookey discloses this limitation. (EX1002 ¶¶121–125.) As discussed, **remote machine 30** (“server computer”) communicates with **client machine 10** over a TCP connection. (*Supra* Section IX.A.1.c.) As explained below, when the machines are communicating over a TCP connection, **remote machine 30** and **client machine 10** would necessarily perform a three-way handshake in order to set up the TCP connection, which would involve the three “communicate” processes recited in this limitation 25(d). (EX1002 ¶121.)

A three-way handshake method is necessarily performed to establish a TCP connection. (*Id.* ¶122; *see also id.* ¶¶46–47.) For example, RFC 793 (EX1008), the TCP specification, provides that a three-way handshake is a necessary procedure used to establish a TCP connection. (EX1008, 32, 34; *id.*, 31, 34–37.) The three-way handshake was known to involve transmitting three messages to negotiate and start a TCP-based session between two devices. (EX1002 ¶¶123; EX1008, 31–32; 34–37.) The ’565 patent acknowledges such understandings, citing RFC 793. (EX1001, 14:26–39.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

The *first* communicated message (step 1) includes a segment with a control bit for synchronization (SYN bit) and thus discloses “communicat[ing] a segment including at least one first synchronize bit,” as claimed. (EX1008, 31; EX1001, 14:26–39.) The *second* communicated message (steps 2 and 3 combined) includes an acknowledgment (ACK) to the segment with the SYN bit and a second SYN bit, and thus discloses “communicat[ing] a first acknowledgement of the segment, and at least one second synchronize bit,” as claimed. (EX1008, 31–32; EX1001, 14:34–35, 14:37–39.) The *third* communicated message (step 4) includes a second ACK to the second SYN bit and thus discloses “communicat[ing] a second acknowledgement,” as claimed. (EX1008, 31–32; EX1001, 14:36–39.)

When operating in accordance with the first protocol to set up a TCP connection between **remote machine 30** and **client machine 10**, **remote machine 30** necessarily performs a three-way TCP handshake that includes the same features recited in this limitation because this handshake was/is a *required* procedure for establishing a TCP connection. (EX1002 ¶¶124–125.) Indeed, PO relies on the

Petition for *Inter Partes* Review
Patent No. 10,075,565

same three-way handshake disclosures in RFC 793 for its allegations regarding this limitation. (EX1017, 84–85.)⁸

e) **25.e**

(1) said server computer further configured to communicate, to a client computer, code that, when used by the client computer, causes the client computer to operate in accordance with a second protocol that is different from the TCP [], in order to setup a second protocol connection with another server computer, by:⁹

The *Wookey-Berg* combination discloses and/or suggests limitation 25.e(1). (EX1002 ¶¶126–161.) As explained, **remote machine 30** (“server computer”) is configured to respond to a request (e.g., request 282) from **client machine 10**

⁸ Petitioner’s references herein to PO’s infringement allegations are not indicative of any concession that any accused instrumentality infringes any claim limitation as alleged by PO.

⁹ The ’565 patent specification does not describe the claimed “code” features as recited in the challenged claims. (EX1002 n.7.) Therefore, while Petitioner addresses this limitation with respect to the prior art, Petitioner does not concede that the ’565 patent provides adequate disclosures of such claimed features.

Petition for *Inter Partes* Review
Patent No. 10,075,565

(“client computer”)¹⁰ by communicating an HTML page 288, that contains encoded URLs, to **client machine 10**. (*Supra* Section IX.A.1.b; EX1002 ¶127.)

The HTML page 288 includes encoded URLs associated with icons (e.g., 257, 257’) on the HTML page 288, and both (1) the encoded URLs included in the HTML page or (2) the HTML page including the encoded URLs disclose the claimed “code.” (EX1002 ¶128.) For example, “[e]ach icon 257, 257’ is associated with *an encoded URL that specifies*: the location of the resource ...; *a launch command associated with the resource*; and *a template identifying how the results of accessing the resource should be displayed.*” (EX1005 ¶[0216].) Also each encoded URL “*contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (*Id.*) Thus, when a user *clicks* an icon 257, 257’ corresponding to a resource from the displayed HTML page 288, **client machine 10** uses the encoded URL associated with the clicked icon to

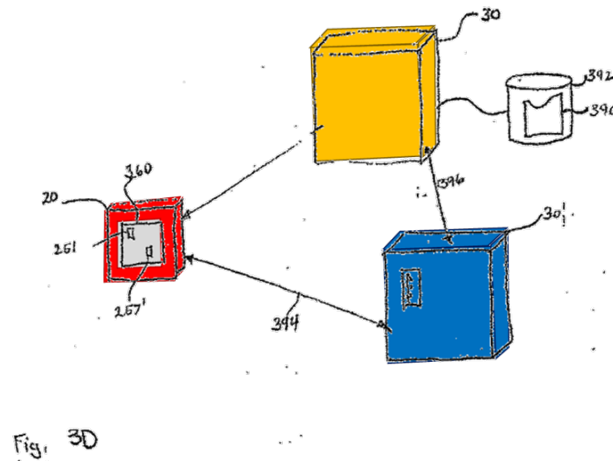
¹⁰ Figures 1A–1B depict block diagrams of typical computer architectures for both remote machines and client machines (e.g., **client machine 10**) (EX1005 ¶[0021]), including personal computers and wireless devices (*id.* ¶¶[0171]–[0173]). Thus, **client machine 10** is a computer for similar reasons discussed above. (*Supra* Section IX.A.1.b; EX1002 ¶127 n.8.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

initiate the process for establishing a second connection (e.g., connection 394 in Figure 3D) with a second **remote machine 30'** ("another server computer")¹¹. (EX1002 ¶129; *see also* EX1005 ¶¶[0026], [0213]–[0216], FIG. 3D.) Accordingly, the encoded URLs are the claimed "code" because such encoded URLs are instructions that, when used by **client machine 10**, command **client machine 10** to access resources from one or more locations (e.g., one or more remote machines). (EX1002 ¶130.)

¹¹ **Remote machine 30'** is a computer for similar reasons discussed above regarding **remote machine 30** and is a server because it provides access to requested resources to **client machine 10** as explained below. (*Supra* Section IX.A.1.b; EX1005 ¶[0021]; EX1002 ¶129 n.9.)

Petition for *Inter Partes* Review
 Patent No. 10,075,565



(EX1005, FIG. 3D (annotated); EX1002 ¶129.)

The above understanding is consistent with how the '565 patent describes its “code” as “an instruction.” (EX1001, cl. 27 (27:44); EX1002 ¶131.) *Wookey*’s encoded URLs similarly include “an instruction,” such as the launch command to connect to the resource. (EX1002 ¶131.)

Moreover, the entire HTML page 288 (including the encoded URLs and icons) also discloses the claimed “code ... used by the client computer,” because an HTML page was known to comprise HTML code used to define, e.g., what is displayed on the screen of the client node. (EX1002 ¶132, ¶¶61–62; EX1012 ¶¶[0022], [0024], [0026], [0079].) The HTML code used to display information relating to page 288 would include the icons associated with the encoded URLs, and when **client machine 10** uses one of the encoded URLs, it causes **client machine**

Petition for *Inter Partes* Review
Patent No. 10,075,565

10 to operate in accordance with a second protocol different from the TCP to set up a second protocol connection with another server (e.g., **remote machine 30'**). PO similarly relates the claimed code to “HTML pages” in its infringement assertions for this and similar limitations. (EX1017, 5 (“to communicate code (e.g., HTML pages, etc.) to the client computer”).)

Thus, *Wookey* discloses, “said server computer further configured to communicate, to a client computer, code that, when used by the client computer, causes the client computer to operate” such that a process for establishing a second connection (e.g., connection 394 in Figure 3D) with a second **remote machine 30'** is initiated.¹² (EX1002 ¶133.)

Regarding the connection established between machines **10** and **30'**, **client machine 10** includes an “HTTP client agent,” such as a web browser application, which “can use any type of protocol.” (EX1005 ¶[0159]; *see also id.* ¶¶[0155], [0192]–[0195].) For example, connection 394 can be made using various protocols, including the Virtual Network Computing (VNC) protocol. (*Id.* ¶[0216].) VNC is

¹² Petitioner’s analysis regarding the “code” limitation equally applies whether the “code” is the encoded URLs in the HTML page or the HTML page itself (including the encoded URLs). (EX1002 ¶133 n.11.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

a desktop-sharing system that allows a computer to remotely access and control another computer. (EX1002 ¶¶63, 134; EX1014 ¶¶[0019].) VNC can run over various “industry standard transport protocols, such as TCP/IP, IPX/SPX, NetBEUI.” (EX1005 ¶¶[0225]; *see also id.* ¶¶[0155], [0215]–[0216].) IPX/SPX and NetBEUI protocols, being non-TCP–based transport protocols, are examples of protocols that are different from the TCP protocol. (EX1002 ¶¶135, 54; EX1013, 1:20–33.)

Wookey further explains that **client machine 10** may use “a different type of protocol than the one used to send the request to the remote machine 30.” (EX1005 ¶¶[0732]; *see also id.*, (“the client machine 10 may make a request to the remote machine 30 using the IPX protocol and request the address of the remote machine 30’ as a TCP/IP protocol address”); EX1002 ¶137.)

Wookey also identifies desired characteristics associated at least with the second connection that a POSITA would have considered when choosing a protocol, such as negotiating connection parameters, reducing unintentional session terminations due to an imperfect connection, detecting/handling disconnections like when a mobile device enters an elevator, and ability to specify an inactive time before connection termination. (EX1002 ¶138; EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153].)

Petition for *Inter Partes* Review
Patent No. 10,075,565

Thus, **client machine 10** identifies negotiable parameters for the connection to be established. (EX1002 ¶¶139–140.) For example, **client machine 10** and **remote machine 30'** “exchange a set of messages which negotiate the parameters under which communications will occur.” (EX1005 ¶[0774].) Once negotiations are complete, **client machine 10** and **remote machine 30'** can communicate as necessary over a network application. (*Id.*; *see also id.* ¶¶[0738]–[0739] (describing the **remote machine 30'** listening for connection requests and processing them), [0744], [0772]–[0773].)

So when **client machine 10** uses one of the encoded URLs (e.g., the one pointing to **remote machine 30'**) associated with one of the icons on the HTML page 288, it results in **client machine 10** utilizing, e.g., the web browser application 280, to establish a connection (e.g., connection 394) with **remote machine 30'** in accordance with a transport protocol. (EX1002 ¶141.)

Once the connection between **client machine 10** and **remote machine 30'** is established (EX1005 ¶[0216]), **remote machine 30'**, which hosts the requested resource, can exchange information with **client machine 10** regarding access to the desired resource. (*Id.*) *Wookey's* **remote machine 30'**, which connects with **client machine 10** to provide access to the desired resource, operates as a server.

Petition for *Inter Partes* Review
Patent No. 10,075,565

(EX1002 ¶142; *see also* EX1005 ¶¶[0150], [0160].) Thus, **remote machine 30'** is the “another server computer,” as claimed. (EX1002 ¶142.)

While *Wookey* discloses various protocols (including protocols that are different from TCP (e.g., IPX)) may be used to establish the second connection and the desired characteristics of such a connection (EX1005, ¶[0225]), *Wookey* does not expressly provide details regarding establishing and maintaining the second connection or the types of negotiable parameters used with those protocols for the second connection. (EX1002 ¶143.) However, it would have been obvious to a POSITA to consider and implement a protocol following *Wookey's* desired characteristics (e.g., such as those identified above) for the second connection between **client machine 10** and **remote machine 30'**, which may be “a different type of protocol than the one used to send the request to the remote machine 30.” (EX1005 ¶[0732].)

Berg describes such a protocol (a “*protocol that is different from the TCP*”) that allows nodes to negotiate or modify timeout timers on a per-connection basis to provide a reliable and efficient network connection. Accordingly, based on the teachings of *Berg*, a POSITA's knowledge, and guidance from *Wookey*, it would have been obvious to configure *Wookey's* **client machine 10** such that when the “code” (e.g., the encoded URL(s) in HTML page 288 pointing to a resource on

Petition for *Inter Partes* Review
Patent No. 10,075,565

remote machine 30', or the HTML page 288 including such URL(s)) is used, the code causes **client machine 10** (e.g., by way of browser application 280) to operate per a second protocol such as the one described by *Berg* (which would have been different than the protocol used to send the request to **remote machine 30**) having at least some characteristics contemplated by *Wookey* for such a second connection between **client machine 10** and **remote machine 30'**, to set up a second protocol connection with **remote machine 30'**. (EX1002 ¶144.)

(a) *Berg*

Berg, like *Wookey*, relates to establishing a reliable connection between a client and server as it states “[t]o provide a reliable backhaul, it is important to be able to maintain multiple IP connections or sessions between the signaling terminal device and the call processing device, so that message[s] can be transmitted even when the inherently unreliable IP network fails” and “[t]he system must be configured so that communications are not interrupted upon network failure, and so that communications can resume when the network comes back up.” (EX1007, 2:4–11; *see also id.*, 1:16–32, 1:54–64, 1:65–2:3, 2:43–47; EX1002 ¶¶81–94 (*Berg* overview).) Thus, *Berg* discloses networking features in the same technical field as *Wookey*. (*See, e.g.*, EX1005 ¶¶[0581] (*Wookey* disclosing that traversal of network segments by client machine 10 may cause changes in its network address or host

Petition for *Inter Partes* Review
Patent No. 10,075,565

name or causes it “to disconnect”), [1135]–[1136] (discussing the need for connections to manage unintentional disconnection between the client and remote machines), [1153] (**client machine 10** losing connection because the user enters an elevator); *id.* ¶¶[0013], [0196], FIGS. 1, 2C, 3D.) A POSITA, therefore, would have had reason to consider *Berg*’s teachings when contemplating and implementing *Wookey*’s method for establishing a second protocol connection between **remote machine 30**’ and **client machine 10**. (EX1002 ¶¶145–146.)

Berg provides a gateway device (acting as a client) connected to a media gateway controller (acting as a server). (EX1007, 3:14–23, 2:55–60.) In *Berg*’s system, both the client and server devices execute a session manager that manages the data network communication session. (*Id.*, 2:54–60, 7:63–8:2, 8:24–31; EX1002 ¶¶147, 81–82.) The session manager “operates logically above a *reliable communication layer*” (EX1007, 2:54–60), which “determines when or if a session is connected or failed” (*id.*). *Berg* explains “[a] protocol layer is ‘reliable’ when it guarantees that it will deliver or generate an error message for every message that an application program requests the transport layer using that protocol to transport.” (*Id.*, 8:27–31; *see also id.*, 8:11–13.) *Berg*’s Figure 2 below describes the layers that computer systems use to communicate over a network, which is based on the Open

Systems Interconnection (OSI) reference model. (*Id.*, 8:3–17, 5:27–55; EX1002 ¶¶147, 22–35.)

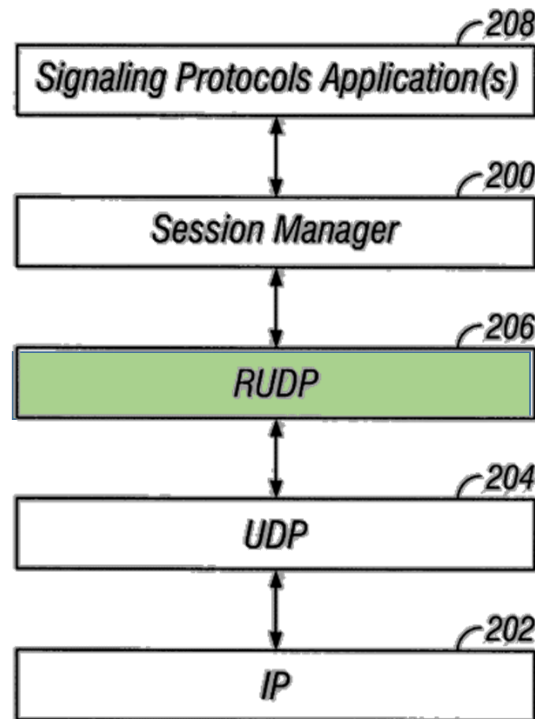


FIG. 2

(EX1007, FIG. 2 (annotated); EX1002 ¶¶147.)

The **Reliable User Datagram Protocol (RUDP) 206** runs on top of the User Datagram Protocol (UDP) protocol software 204 at a transport layer. (EX1007, 8:10–17, 16:66–17–17; EX1002 ¶¶148, 56–60.) The UDP is a non-TCP communications protocol (e.g., a protocol that is different from the TCP) that is primarily used for establishing low-latency and loss-tolerating connections, especially time-sensitive transmissions, between applications. (EX1002 ¶¶148, 55;

Petition for *Inter Partes* Review
Patent No. 10,075,565

EX1015 ¶[0110]; EX1016 ¶[0010].) The RUDP layer 206 allows *Berg*'s system to determine “when or if a session is connected or failed.” (EX1007, 2:54–60.)¹³ Above the **RUDP layer 206** is the Session Manager 200 and Signaling Software Application(s) 208. (*Id.*, 8:24–27.)

The Session Manager 200 can run above any reliable communication mechanism (e.g., RUDP 206). (*Id.*) A communication layer “is ‘reliable’ when it guarantees that it will deliver or generate an error message for every message that an application program requests the transport layer using that protocol to transport.” (*Id.*, 8:27–31; *see also id.*, 8:11–13; EX1002 ¶149.)

Berg notes that the “TCP/IP has a number of characteristics that make it unsuitable for” some applications. (*Id.*, 17:28–30.) In particular, “[m]ost TCP/IP implementations that allow properties like timers to be modified, *do not allow* the modification to be done on a per-connection basis.” (*Id.*, 17:40–42; EX1002 ¶150.)

Contrasted to TCP/IP implementations, “RUDP is designed *to allow characteristics of each connection to be individually configured* so that many

¹³ *Berg* uses “RUDP layer 206” and “communication layer” to refer to the same layer in the exemplary OSI model in Figure 2. (EX1002 ¶148 n.14; EX1007, 2:54–60, 8:13, 8:24–28.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

protocols with different transport requirements can be implemented simultaneously on the same platform.” (EX1007, 17:42–46.) For example, RUDP allows the nodes to negotiate various parameters including the timeout values associated with timers for the retransmission timeout (*id.*, 20:24–29, 22:44–56), acknowledgement timeout (*id.*, 20:29–35, 23:15–24), null segment (*id.*, 20:36–41, 23:45–59), and transfer state (*id.*, 20:42–47, 24:15–25). *Berg* explains that the messages sent to set up the RUDP connection are UDP packets that include an RUDP header. (*Id.*, 17:59–18:20, 19:1–40; EX1002 ¶151; *see also infra* Section IX.A.1.c.)

The “RUDP is a lightweight protocol layer designed to run on top of UDP [that] can provide reliable in-order delivery” (EX1007, 16:66–67; *see also id.*, 17:9–16) and it “has a *very flexible design* that would make it *suitable for a variety of transport uses*. (*Id.*, 17:13–15; EX1002 ¶152.)

Thus, the RUDP is “a second protocol that is different from the TCP” (e.g., a non-TCP protocol), and operating in accordance with the RUDP would have caused a client to set up an RUDP connection (“a second protocol connection”) with a server. (EX1002 ¶153.) *See also* Unified -742 IPR, Paper 11 at 10 (Board agreeing that *Berg*’s RUDP is a non-TCP protocol).

(b) Reasons to Combine

Based on *Wookey*'s and *Berg*'s disclosures and a POSITA's knowledge at the time, it would have been obvious to configure and implement *Wookey*'s **client machine 10** (including its web browser application 280) to use a protocol like *Berg*'s RUDP ("a protocol that is different from the TCP") in order to set up an RUDP ("second protocol") connection with **remote machine 30'** (*see, e.g.*, EX1005, Figure 3D) in relation to claim limitations 25.e–h and claim 28. (EX1002 ¶¶154–161.) Such an implementation would have had many advantages and been a predictable and straightforward result achieved by combining well-known technologies using known methods. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 416–18 (2007).

As explained, both *Wookey* and *Berg* are directed to establishing a reliable connection between two nodes, where one or both nodes may experience an unintentional disconnection, and thus disclose features in a similar technological field. (EX1002 ¶¶155, 20–21; *see supra* Section IX.A.1.e.1.a; EX1005 ¶¶[0581], [1135]–[1136], [1153]; EX1007, 1:16–32, 1:54–2:11, 2:43–47 (discussing importance of maintaining connections and handling disruptions).) Thus, a POSITA would have had reason to consider *Berg* when contemplating and implementing the teachings of *Wookey*. (EX1002 ¶¶154–155.) And when collectively considered, such a POSITA would have been motivated to modify *Wookey*'s **client machine 10**

Petition for *Inter Partes* Review
Patent No. 10,075,565

to use a protocol like *Berg*'s RUDP in order to set up an RUDP connection with **remote machine 30'**.

Indeed, as discussed, *Wookey* describes using "code" to cause **client machine 10**, e.g., a web browser application 280, to use a protocol that can be different from the one used to connect **client machine 10** and **remote machine 30**, in order to set up a second protocol connection between **client machine 10** and **remote machine 30'**. *Wookey* also describes the types of characteristics concerning the protocol that can be used for the second connection, such as negotiating connection parameters, reducing unintentional terminations, detecting/handling disconnections, specifying a permissible inactive time before connection termination. (EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153]; *supra* Section IX.A.1.b.1.) A POSITA would have been thus motivated by *Wookey*'s disclosures/suggestions to consider and incorporate a protocol that would provide the desired features to *Wookey* for the second protocol connection with **remote machine 30'**, such as the RUDP described by *Berg*. (EX1002 ¶156.)

A POSITA would have recognized the benefits of using the RUDP protocol described in *Berg* with *Wookey*'s system/method, such as providing **client machine 10** with an improved reliable connection to **remote machine 30'** with the versatility of negotiated timeout parameters consistent with *Wookey*'s desired characteristics

Petition for *Inter Partes* Review
Patent No. 10,075,565

for the second connection. *See KSR*, 550 U.S. 416–17. Indeed, by using a protocol like *Berg*’s RUDP protocol, *Wookey*’s method would enable negotiation of properties like the timeout values on a per-connection basis over a reliable connection (EX1007, 24:40–47), and provide alerts to the nodes if the session fails (*id.*, 2:55–60). (EX1002 ¶157.) *Wookey*’s method for “providing a client with a reliable connection to a host service” (EX1005 ¶[0077]) would therefore have been improved by providing a connection that is “very flexible ... [and] suitable for a variety of transport uses,” as provided by *Berg* (EX1007, 17:9–16). (EX1002 ¶157; EX1005 ¶[0903].)

It would have been a foreseeable and straightforward implementation to configure *Wookey*’s **client machine 10** such that the “code” (e.g., the encoded URLs in HTML page 288 or the HTML page 288 including such URLs), when used, would cause **client machine 10** (e.g., by way of its web browser 280) to operate in accordance with a second protocol different from the TCP like *Berg*’s RUDP. (EX1002 ¶158.) *See KSR*, 550 U.S. at 416. For example, as noted above, “*the URL includes a file, or a reference to a file, that contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (EX1005 ¶[0216].) In light of the reasons discussed above, a POSITA would have been motivated to configure *Wookey*’s encoded URL such that it

Petition for *Inter Partes* Review
Patent No. 10,075,565

includes information causing the client to establish a connection with **remote machine 30'** using a second protocol different from the TCP (like *Berg's* RUDP). (EX1002 ¶158.) This implementation would have been achieved by combining well-known technologies using known methods, such as known network design concepts and technologies described above by *Wookey* and *Berg*, and known in the art at the time. (*Id.* ¶158.)

Given that *Berg's* protocol is flexible and suited for a variety of uses and allows the nodes to negotiate properties like the timeout values on a per-connection basis over a reliable connection (EX1007, 24:40–47), a POSITA would have considered these features when implementing *Wookey*, something *Wookey* itself recognizes (EX1005 ¶[0721]), and are features that would have improved *Wookey's* process just as described in *Berg*. (EX1002 ¶159.) Accordingly, such a combination would have yielded the predictable result of providing a network communication exchange that minimizes disruptions caused by timeout issues. (*Id.* ¶159); *see KSR*, 550 U.S. at 417.

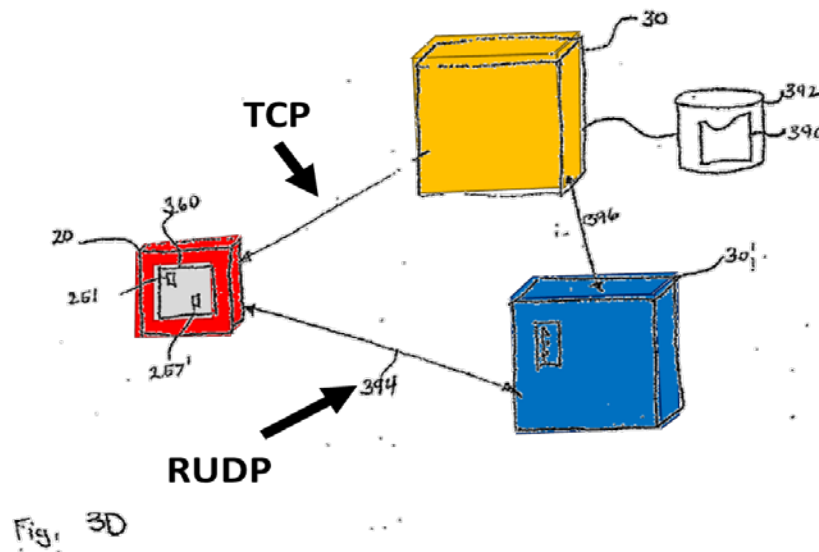
Additionally, the *Wookey-Berg* combination would have involved the use of known technologies (e.g., aspects of similar protocols) and design concepts and processes to obtain the foreseeable result of a reliable connection between *Wookey's* **client machine 10** and **remote machine 30'** using web browser application 280.

Petition for *Inter Partes* Review
Patent No. 10,075,565

(EX1002 ¶160.) Indeed, the above-modification would have involved the substitution of features from one reliable protocol from a finite number of available alternative reliable communication protocols, such as that described by *Berg*. (*Id.* ¶160.) Thus, a POSITA would have had a reasonable expectation of success in the above-modification. (*Id.* ¶160.) See *Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1364 (Fed. Cir. 2007) (“only a reasonable expectation of success, not a guarantee, is needed” in an obviousness analysis).¹⁴

A POSITA would have been skilled and knowledgeable to configure the above modification in various ways, while taking into account any known programming, design, and other related concepts, limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. (EX1002 ¶160.) For example, a POSITA would have been motivated based on such disclosures to configure *Wookey*’s system and process consistent with the non-limiting example reflected below.

¹⁴ There is no evidence of nonobviousness at this time that would overcome the overwhelming evidence of obviousness as presented herein. Petitioner reserves the right to address any such evidence/argument should PO later present any in this proceeding.



(EX1005, FIG. 3D (annotated); EX1002 ¶160.)

* * *

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 25.e(1). (*Id.* ¶161; *see also infra* Section IX.A.1.b.2, IX.A.1.c–e.)

(2) [a second protocol that is different from the TCP and] that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer,

The *Wookey-Berg* combination discloses and/or suggests and renders obvious limitation 25.e(2). (EX1002 ¶¶162–169.) As explained above, the code in the *Wookey-Berg* combination causes **client machine 10** (e.g., by way of its web browser application 280) to operate in accordance with the RUDP when establishing

Petition for *Inter Partes* Review
Patent No. 10,075,565

a connection between **client machine 10** and **remote machine 30**'. (*Supra* Section IX.A.1.e.1; EX1002 ¶163.) As discussed below, the *Wookey-Berg* combination discloses and/or suggests that the RUDP ("a second protocol that is different from the TCP") would have operated above an Internet Protocol (IP) layer and below a Hypertext Transfer Protocol (HTTP) application layer, as claimed.

For example, "a user of the client machine 10 can access a resource by clicking an icon 257, 257' displayed in the Resource Neighborhood web page" (EX1005 ¶[0216]), which is executed via "a web browser application 280" (*id.* ¶[0192]) over the World Wide Web. (EX1002 ¶164.) **Client machine 10**'s web browser application 280 initiates requests according to HTTP (EX1005 ¶[0155]), which was a known communication protocol at the application layer of an OSI model or TCP/IP model¹⁵, where hypertext documents include hyperlinks to other resources the user can easily access via a web browser. (EX1002 ¶¶164, 30; EX1018, 7–8, 12.)

¹⁵ For purposes of this IPR, regardless of the model (OSI or TCP/IP) considered at the time, a POSITA would have had the same understanding as explained herein. (EX1002 ¶164 n.16.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

Indeed, *Wookey* provides that the communications link between the remote and client machines may use an HTTP as the application layer protocol. (EX1005 ¶[0155].) *Wookey* also explains that **client machine 10** includes an “HTTP client agent,” such as a web browser, which “can use any type of protocol.” (*Id.* ¶[0159]; EX1002 ¶165.)

The Reliable User Datagram Protocol (RUDP) in the *Wookey-Berg* combination would have operated below the application layer on which the HTTP operates. (EX1002 ¶166.) This is because, in the *Wookey-Berg* combination, the RUDP would have run on top of the User Datagram Protocol (UDP) protocol software 204 at a transport layer, and information exchanged via the RUDP between the client and server would have been communicated to the application layer where the HTTP-based web browser operated. (*Id.*) This transport layer is below the application layer. (*Id.*) Similarly, the RUDP in the *Wookey-Berg* combination would have operated above an IP layer because the transport layer operates above the IP layer. (*Id.*)

Berg confirms this understanding. Specifically, *Berg* discusses the logical layers that computer systems use to communicate over a network in relation to its use of RUDP concerning Figure 2, as discussed above. (EX1007, 8:3–8; *see supra* Section IX.A.1.b.1.a.) As shown below concerning Figure 2, the **RUDP layer 206**

operates above **IP layer 202** and below **Signaling Protocols Application(s) layer 208** (an application layer, where the HTTP would operate, i.e., “a hypertext transfer protocol (HTTP) application layer”). (EX1007, FIG. 2, 8:3–18, 8:24–31; *see also id.*, 1:15-59, 5:27–55, EX1002 ¶167.)

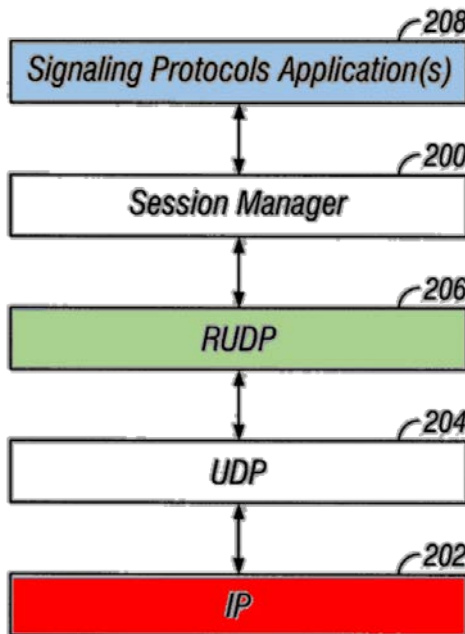


FIG. 2

(EX1007, FIG. 2 (annotated); EX1002 ¶167.)

Thus, the combination of *Wookey-Berg* discloses and/or suggests and renders obvious that the second protocol that is different from the TCP “operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer,” as claimed in limitation 25.e. (EX1002 ¶¶168–169.)

f) receiving, by the client computer from the another server computer, a packet;

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶170–174.) As discussed above, the *Wookey-Berg* combination would have been configured such that **client machine 10** establishes an RUDP connection, according to the RUDP protocol (“a second protocol that is different from the TCP”), between **client machine 10** and **remote machine 30’**. (*Supra* Section IX.A.1.e.) As explained below, the “code” in the combined *Wookey-Berg* system/method would, when used by **client machine 10**, results in **client machine 10** using a web browser application 280 to operate in accordance with the RUDP protocol to receive by **client machine 10** (“client computer”) from **remote machine 30’** (“another server computer”) an SYN segment (“packet”) during the setup of the RUDP connection to allow the nodes to negotiate parameters for the connection as discussed by *Berg*. (*E.g.*, EX1007, 18:56–62, 24:39–47; EX1002 ¶174.)

For example, “RUDP is designed to allow characteristics of each connection to be individually configured.” (EX1007, 17:42–46.) Thus, when a client initiates a connection, it would send “a SYN segment which contains the negotiable parameters defined by the [Upper Layer Protocol] ULP via the [Application Programmer Interface] API.” (*Id.*, 24:40–42; *see also id.*, 18:56–62; EX1002 ¶171.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

The server, upon receiving the client's SYN segment, determines to either "accept" the proposed parameters in the received SYN segment or propose different parameters. (EX1007, 24:42–44.) In either case, the server returns the parameters in its responsive SYN segment to the client. (*Id.*) The client, upon receipt of the server's SYN segment, can then choose to accept the parameters sent by the server by responding to the server with an ACK message to establish the connection, or it can send a reset (RST) segment to the server to refuse the connection. (*Id.*, 24:44–47; EX1002 ¶172.)

Berg explains that the messages sent to set up the RUDP connection, including the SYN segments, are UDP packets that include an RUDP header. (EX1007, 17:59–18:20; *see also id.*, 19:1–40 (SYN segment); EX1002 ¶173.) Therefore, the SYN segment response sent from **remote machine 30'** ("another server computer") and received by **client machine 10** ("client computer") in the *Wookey-Berg* combination is a packet (e.g., a UDP packet with an RUDP header). (EX1002 ¶¶173–174.)

g) 25.g

(1) **identifying metadata, that specifies a number of seconds or minutes, in an idle time period parameter field in the packet for an idle time period, []**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶175–191.) During the setup of the RUDP connection between **client machine 10** and second **remote machine 30'** in the above-disclosed combined *Wookey-Berg* system/method, **client machine 10** would receive a SYN segment (“packet”) from **remote machine 30'** to allow the nodes to negotiate parameters consistent with the features disclosed by *Berg*. (*Supra* Section IX.A.1.c; EX1002 ¶176.) As explained below, the combined *Wookey-Berg* system/method would have been configured such that **client machine 10** identifies various negotiable parameters from the parameter fields in the SYN segment (“packet”), including a timeout value (“metadata”) that specifies a number of seconds for an idle time period. (EX1002 ¶176.)

A POSITA implementing the above-discussed combination would have recognized that *Berg* discloses identifying the negotiable parameters in the SYN segment (“packet”) for establishing an RUDP connection. (EX1002 ¶177.) For example, when the client receives the SYN segment sent from the server, it can *accept* or *reject* the proposed parameters (such as the null segment timeout value) in

Petition for *Inter Partes* Review
Patent No. 10,075,565

the received SYN segment. (EX1007, 24:44–48.) Thus, the client necessarily identifies each parameter in the SYN segment to decide whether it will accept or reject the proposed parameters. (EX1002 ¶177.) The *Wookey-Berg* combination would have thus been configured such that **client machine 10** identifies data (“metadata”) in the parameter fields in the RUDP SYN segment (“packet”). (*Id.* ¶¶177–178.)

To the extent the *Wookey-Berg* does not disclose the claimed “identifying,” it would have been obvious to modify the functionality in the combined *Wookey-Berg* system/process such that it identifies the negotiable parameters like that claimed. (*Id.* ¶179.) A POSITA would have been motivated to configure the *Wookey-Berg* system/process to include functionality that identifies the parameters to facilitate the subsequent process of accepting or rejecting the parameter in the SYN segment in a manner consistent with the features described by *Berg*. (EX1007, 24:44–48; EX1002 ¶¶179–180.)

For example, Figure 2, below, of the RUDP specification¹⁶, illustrates the SYN segment with all of its negotiable parameters for the RUDP connection:

¹⁶ The RUDP specification is included in *Berg* as “Appendix 1”. (EX1007, 16:59–25:54; *see also* EX1011, 6; EX1002 ¶181 n.18.)

Petition for *Inter Partes* Review
Patent No. 10,075,565



(EX1007, 19:1–40; EX1002 ¶181.)

The *Wookey-Berg* combination would have thus been configured such that **client machine 10** identifies data (“metadata”) in the parameter fields in the RUDP SYN segment (“packet”). (EX1002 ¶182.)

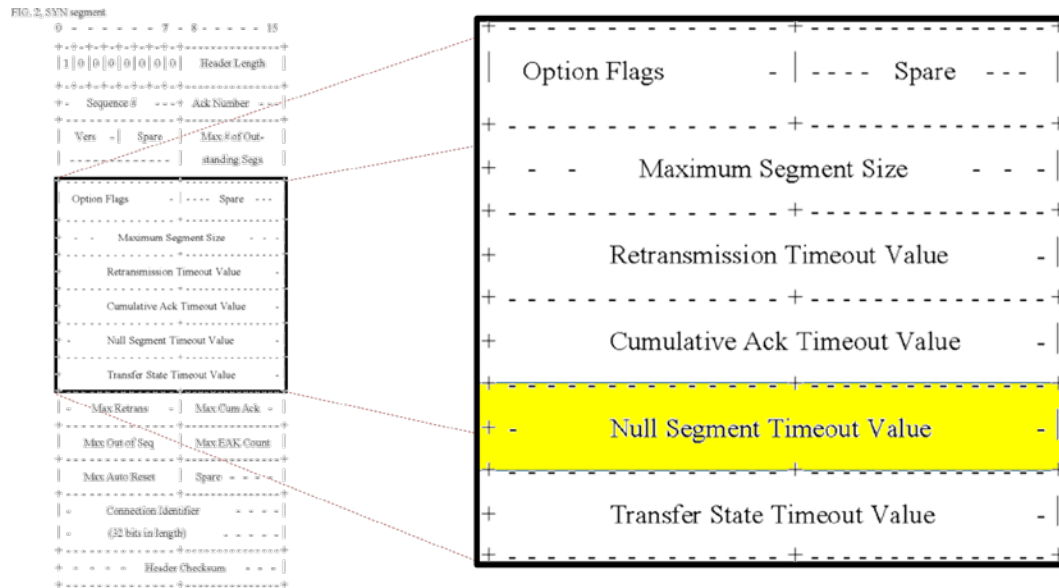
As explained below, the negotiable parameters for (1) the null segment timeout value and (2) transfer state timeout value each separately represent “metadata, that specifies a number of seconds or minutes, in an idle time period parameter field in the packet for an idle time period,” as claimed. (*Id.* ¶¶183–191.)

(a) Null Segment Timeout Value

Berg discloses that a null segment timeout value (“metadata”), which is specified in milliseconds (which would have indicated “a number of seconds,” as claimed), is identified from a null segment timeout value parameter field (“idle time period parameter field”) in an RUDP SYN segment (“packet”). (*E.g.*, EX1007, 18:58–60, 19:22–25, 20:36–42, 22:25–33; *see also* EX1002 ¶¶184–185.) The null segment timeout value indicates the amount of time (*e.g.*, milliseconds) to wait before sending a null segment when the connection has been idle (*e.g.*, when a data segment has not been sent or received). (EX1007, 20:36–40; EX1002 ¶184.)¹⁷ The null segment is sent to determine “if the other side of a connection is still active.” (EX1007, 22:25–33, 20:3–40.) This period of waiting by the client (when the connection is idle) before the client sends a null segment to the server to verify whether the connection is active represents an “idle time period,” as claimed. (EX1002 ¶184; *see also infra* Section IX.A.1.g.2 (explaining the null segment time period represents an idle period).) Figure 2 of the RUDP specification illustrates the

¹⁷ *Berg* uses “null segment” and “NUL segment” interchangeably. (EX1007, 20:36–42, 22:25–32; EX1002 ¶184 n.19.)

null segment timeout value parameter field that specifies the **null segment timeout value**:



(*Id.*, 19:1–40 (excerpted and annotated); *see also* EX1002 ¶184.)

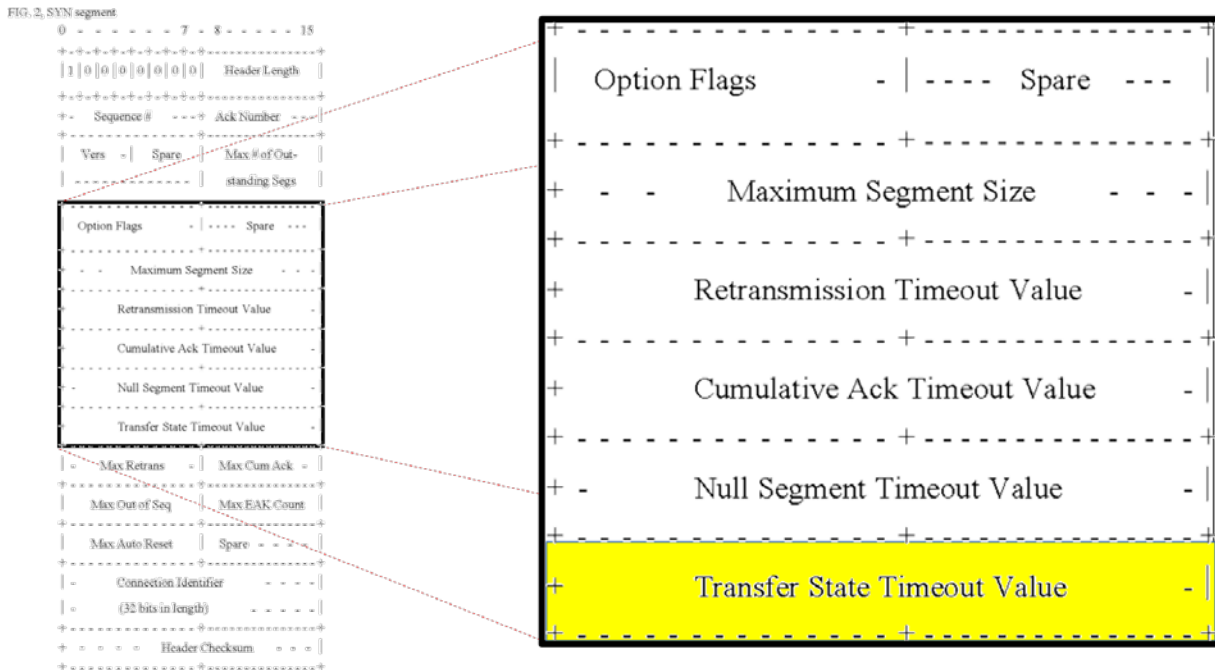
The *Wookey-Berg* combination would have thus been configured such that **client machine 10** identifies data such as the null segment timeout value (“metadata”), that specifies a number of seconds or minutes, in the null segment timeout value parameter field (“idle time period parameter field”) in the RUDP SYN segment (“packet”) for an idle time period, as claimed. (EX1002 ¶185.)

(b) Transfer State Timeout Value

Berg also discloses that a transfer state timeout value (“metadata”), which is specified in a number of seconds, is identified from a transfer state timeout value parameter field (“idle time period parameter field”) in an RUDP SYN segment

Petition for *Inter Partes* Review
 Patent No. 10,075,565

(“packet”). (E.g., EX1007, 19:25–29, 20:36–42, 22:25–33; *see also* EX1002 ¶¶186–191.) Figure 2, below, of the RUDP specification, illustrates the transfer state timeout value parameter field that specifies the **transfer state timeout value**:



(EX1007, 19:1–40 (excerpted and annotated); EX1002 ¶¶186.)

The transfer state timeout value determines, using a transfer state timer (set to the transfer state timeout value), how long a node will wait to transfer the state to another connection before deeming the connection state lost. (EX1007, 24:15–25; *see also id.*, 20:42–47.) When the transfer state timer expires, the connection’s data is freed up, and thus **client machine 10** and **remote machine 30’** are no longer connected via the RUDP connection. (EX1002 ¶¶187–188.) For example, when an RUDP connection fails, the nodes will save the current connection state and close

Petition for *Inter Partes* Review
Patent No. 10,075,565

the current RUDP connection. (EX1007, 24:15–25; EX1002 ¶188.) The nodes will seek to initiate a new connection to transfer the saved state of the previous failed RUDP connection. (EX1007, 24:15–25.) The transfer state timeout value is the negotiated time that either node will wait before deeming the connection state lost. (*Id.*)

The transfer state timer, before it expires, causes the state associated with the failed RUDP connection to be kept at least partially alive while waiting to transfer to another connection. (EX1002 ¶189; EX1007, 24:15–25, 20:42–47) Thus, this period of waiting by the client (when the connection is idle) before the client deems the RUDP connection lost represents an “idle time period.” (EX1002 ¶189.)

The *Wookey-Berg* combination would have thus been configured such that **client machine 10** identifies data such as the transfer state timeout value (“metadata”), that specifies a number of seconds or minutes, in the transfer state timeout value parameter field (“idle time period parameter field”) in the RUDP SYN segment (“packet”) for an idle time period, as claimed. (*Id.* ¶190.)

* * *

The ’565 patent confirms the understanding that the null segment and transfer state timeout values are metadata. (EX1002 ¶191.) For example, the ’565 patent explains that metadata can be a value indicating a “duration of time” (EX1001,

Petition for *Inter Partes* Review
Patent No. 10,075,565

21:34–39), which is what each of the above-discussed timeout values in *Berg* contains. (EX1007, 20:36–47; EX1002 ¶191.)

Moreover, the above understanding regarding the above-discussed timeout values is consistent with PO’s view of the same claimed features. (EX1017, 88 (PO alleging “identify[ing] metadata (e.g., a value, etc.) that specifies a number of seconds or minutes, in an idle time period parameter field (e.g., idle_timeout parameter field, etc.) in the packet for an idle time period”), (“the metadata includes a value in seconds”).)

(2) [an idle period] during which, no packet is communicated that meets each of the following criteria: a) communicated via the second protocol connection, and b) causes the second protocol connection to be kept at least partially alive; and

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶192–208.) As explained above, the combined *Wookey-Berg* system/method would have been configured such that **client machine 10** identifies various negotiable parameters from the parameter fields in the SYN segment (“packet”), including a null segment timeout value and the transfer state timeout value (each individually representing metadata for an “idle period”), which are both specified in a number of seconds. (*Id.* ¶192.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

As explained below, the null segment timeout value and the transfer state timeout value in the combined *Wookey-Berg* system/method would have each specified an idle time period as claimed. (*Id.* ¶193.) Moreover, as explained below, the *Wookey-Berg* combination also discloses this limitation in the scenario where at least one node is disconnected as envisioned by the combined system/method. (*Id.*)

(a) Null Segment timeout Value

As explained below, the client-side null segment timer is configured such that during the idle time period (based on the null segment timeout value), “no packet is communicated that meets each of the following criteria: a) communicated via the second protocol connection, and b) causes the second protocol connection to be kept at least partially alive.” (EX1002 ¶¶194–201.)

The client’s null segment timer is set to the time period specified by the null segment timeout value. (EX1007, 20:36–37 (describing that the null segment timeout value corresponds to sending a null segment), 23:46–49; EX1002 ¶195.) “If the client’s null segment timer expires, the client sends a null segment to the server” in the established RUDP connection. (EX1007, 23:46–48.) The client’s null segment timer “is started when the RUDP connection is opened.” (*Id.*) The timer “is reset every time a data segment is sent.” (*Id.* (emphasis added).) Thus, with respect to *Berg*’s client-side null segment timer, the packet that must be “a)

Petition for *Inter Partes* Review
Patent No. 10,075,565

communicated via the second protocol connection” and which “b) causes the second protocol connection to be kept at least partially alive” is a *data segment* sent by the client. (*Id.*; EX1002 ¶196.)

The RUDP connection is “kept at least partially alive” during the period defined by the null segment timeout value because as explained, the null segment sent by the client in the RUDP connection is used to determine if the other side of a connection is still active. (*Id.*, 22:25–27, 23:45–59; *supra* Section IX.A.1.g.1.a; EX1002 ¶197.) When a node (e.g., the server) receives a null segment, it must acknowledge the segment to verify that a valid connection exists. (EX1007, 22:27–30.) On the other hand, if the null segment is not received or acknowledged by the server, the connection is subject to being closed via an “auto reset.” (*Id.*, 23:45–59; EX1002 ¶197.)

The “auto reset,” which is triggered upon detection of the null segment timer’s expiration, deactivates the RUDP connection between the client and server. (EX1002 ¶198; *see also* EX1007, 21:15–23, 23:60–24:4.) For example, an “auto reset” signals a problem with the connection (e.g., connection failure) and that the client needs to deactivate the RUDP connection. (EX1007, 23:60–24:4; *see also id.*, 21:15–23 (explaining that auto-reset attempts to reconnect the connection, and if the Max Auto Reset is zero, the nodes will not attempt to reconnect), 24:15–25.) When

Petition for *Inter Partes* Review
Patent No. 10,075,565

an auto-reset is necessary, the client will send a reset (RST) segment to close or reset a connection. (*Id.*, 22:20–24, 23:60–24:4; *id.*, 18:27–28, 22:19–24; EX1002 ¶199.) Either closing or resetting a connection is deactivation of the connection because in either case no further data is exchanged between the nodes over that connection. (EX1002 ¶199.) Thus, upon detection of the null segment timer’s expiration, the RUDP connection would have been subject to deactivation (e.g., by way of an auto-reset) such that the RUDP connection would not be kept at least partially active. (*Id.* ¶200)

Accordingly, in the combined *Wookey-Berg* system/method, the null segment timeout value defines an idle time period for the client’s null segment timer during which, no packet (here, a data segment) is communicated that meets each of the following criteria: a) communicated via the RUDP connection (“second protocol connection”), and b) causes the RUDP connection to be kept at least partially alive, as claimed. (*Id.* ¶201.)

(b) Transfer State Timeout Value

Berg explains that the transfer state timeout value indicates the amount of time the state information will be saved for a connection after an auto-reset occurs. (EX1007, 20:43–45; EX1002 ¶¶202–203.) For example, when an RUDP connection fails, each node will save its current connection state. (EX1007, 24:15–25.) Each

Petition for *Inter Partes* Review
Patent No. 10,075,565

node will also seek to initiate a transfer of this saved state to another connection, e.g., a new connection, by communicating a SYN segment to the other node. (*Id.*, 18:65–67, 23:60–24:4; *see also id.*, 24:15–25.) For example, the SYN segment is used to perform an auto-reset on a connection. (*Id.*, 18:65–67.) During this process, a transfer state timer will be set to the transfer state timeout value (defining an “idle time period”). (*Id.*, 24:21–23, *see also id.*, 20:43–45.) If the transfer state timer expires before a new connection is established, the saved queues corresponding to the lost connection are freed up (i.e., removed). (*Id.*, 24:21–23.) Thus, for *Berg*’s transfer state timer, the packet that must be “a) communicated via the second protocol connection” and which “b) causes the second protocol connection to be kept at least partially alive” is a *SYN segment*. (EX1002 ¶202.)

Accordingly, in the combined *Wookey-Berg* system/method, the transfer state timeout value defines an idle time period for the client’s transfer state timer during which, no packet (here, a SYN segment) is communicated that meets each of the following criteria: a) communicated via the RUDP connection (“second protocol connection”), and b) causes the RUDP connection to be kept at least partially alive, as claimed. (*Id.* ¶203.)

Disconnection

Moreover, both idle time periods discussed above (based on the null segment timeout value and the transfer state timeout value) satisfy claim limitation 25.g.2 in an **additional** way by the *Wookey-Berg* combination where a disconnection occurs between the client machine 10 and remote machine 30'.¹⁸ (EX1005 ¶¶[0581], [1135]–[1136], [1153]; EX1007, 2:4–11, 1:16–32, 1:54–64, 1:65–2:3, 2:43–47; *see also supra* Section IX.A.1.e.1.b (noting that both *Wookey* and *Berg* aim to maintain connection between nodes through unintentional disconnection periods); EX1002 ¶¶204–206.) For example, when an interruption in communication occurs, such as a disconnection in the network service (as contemplated by *Wookey* and *Berg*) or in a physical network medium (as described in the '565 patent) (EX1001, 2:5–8, 2:23–26, 12:37–43), packets from one node do not reach the other node in either direction. (EX1002 ¶205.) In this situation contemplated by the '565 patent, in both idle time periods discussed above (i.e., based on the null segment timeout value and transfer state timeout value), no packet is communicated that meets each of the following

¹⁸ This is consistent with how the '565 patent describes such features. (EX1001, 12:34–43, 2:23–26, 21:56–63 (idle time period may be linked to a network medium physical disconnection of a network medium or simply a dead connection).)

Petition for *Inter Partes* Review
Patent No. 10,075,565

criteria: a) communicated via the RUDP connection (“second protocol connection”), and b) causes the RUDP connection to be kept at least partially alive, because no packet can get through the interruption or the disconnection in the physical medium. (*Id.* ¶205; EX1001, 12:34–43, 2:23–26, 21:56–63, FIG. 7 (annotated below in yellow showing no communication between the nodes during a disconnection).)¹⁹ This is the type of situation where, as discussed above, *Wookey* and *Berg* both aim to

¹⁹ The ’565 patent repeatedly confirms that in order for a packet to be “communicated . . . to keep the . . . connection active,” a packet sent by one node must be received by the other node. (EX1002 ¶206; EX1001, 11:10–15 (“With reference to the method illustrated in FIG. 2, block 202 illustrates the method includes receiving, by a first node, first idle information for detecting a **first idle time period during which no TCP packet** including data in a first data stream **sent in the TCP connection by a second node is *received* by the first node.**”) (emphasis added), 11:15–20 (disclosing that during the idle time period, no TCP packet “**sent in the TCP connection by a second node is *received* by the first node**”), 11:20–25 (same), 20:51–21:3 (same), 19:33–40 (disclosing that during the idle time period, “**no TCP packet in the TCP connection is *received***, by the first node 602 . . . from second node 604”), FIG. 2 (block 202), FIG. 3 (block 304), FIGS. 6–7.)

Petition for *Inter Partes* Review
 Patent No. 10,075,565

maintain the connection, e.g., maintain the connection even when no packet is communicated during the negotiated null segment timeout value used to set the null segment timers or during the negotiated transfer state timeout value used to set the transfer state timer. *See Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016) (“[C]ombinations of prior art that sometimes meet the claim elements are sufficient to show obviousness.”); *Power Integrations, Inc. v. Fairchild Semiconductor Int'l, Inc.*, 843 F.3d 1315, 1336-37 (Fed. Cir. 2016) (same).

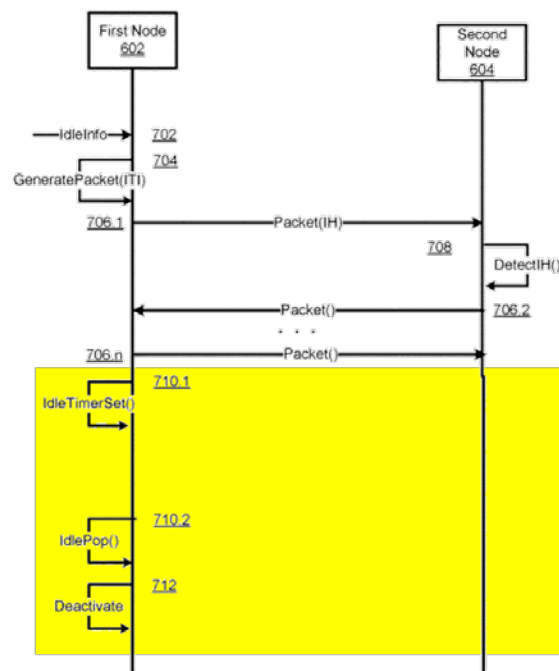


Fig. 7

(EX1001, FIG. 7 (annotated); EX1002 ¶205.)

* * *

Petition for *Inter Partes* Review
Patent No. 10,075,565

Accordingly, for reasons similar to those discussed above for claim limitations 25.e–f, a POSITA would have found it obvious to incorporate the above features in the *Wookey-Berg* combination. (EX1002 ¶207; *supra* Sections IX.A.1.e–f (discussing and illustrating the *Wookey-Berg* combination).) For instance, in the above-discussed *Wookey-Berg* combination, a POSITA would have been motivated and found obvious to implement in the process of establishing the connection between **client machine 10** and **remote machine 30** to include a process of **client machine 10** identifying data such as the null segment timeout value or transfer state timeout value (“metadata”), that specifies a number of seconds or minutes, in the null segment timeout value parameter field or the transfer state timeout value parameter field (“idle time period parameter field”) in the RUDP SYN segment (“packet”) for an time period specified by the null segment timeout value and the transfer state timeout value (both separately disclosing an “idle time period”), during which no packet is communicated that meets each of the following criteria: a) communicated via the RUDP connection (“second protocol connection”), and b) causes the RUDP connection to be kept at least partially alive, as claimed. (*See supra* Section IX.A.1.e.) Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 25.g. (EX1002 ¶208.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

h) determining, by the client computer and based on the metadata, a timeout attribute associated with the second protocol connection.

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶209–215.) As explained above for limitation 24.g, **client machine 10** in the *Wookey-Berg* combination identifies various negotiable parameters in the SYN segment, including the null segment timeout value (“metadata”) and transfer state timeout value (“metadata”) for establishing the RUDP connection (“second protocol connection”). (*Supra* Section IX.A.1.g.) As explained above and further explained below, **client machine 10** in the *Wookey-Berg* combination would have also been configured to determine timeout attributes, such as a null segment timer (“timeout attribute”) and transfer state timer (“timeout attribute”), based on the parameters (“metadata”)—e.g., the null segment timeout and transfer state timeout values—associated with the RUDP connection (“second protocol connection”). (EX1002 ¶209; *see supra* Section IX.A.1.g.)

For example, the client determines the timeout value for a null segment timer (“timeout attribute”) associated with the RUDP (“second protocol”) connection based on the null segment timeout value (“metadata”). (EX1007, 20:36–38, 23:46–59; EX1002 ¶210; *see supra* Section IX.A.1.g.) As discussed above, the null segment timeout value is at least based on the received SYN segment. (*Supra*

Petition for *Inter Partes* Review
Patent No. 10,075,565

Section IX.A.1.g.1.a.) The client's null segment timer is set based on the null segment timeout value (e.g., when the client accepts the null segment timeout value proposed by the server). (EX1007, 23:45–59; *see also id.*, 20:36–41; EX1002 ¶210; *supra* Section IX.A.1.g.2.a.) *See, e.g., Unwired Planet, LLC*, 841 F.3d at 1002; *Power Integrations, Inc.*, 843 F.3d at 1336-37. Accordingly, in the *Wookey-Berg* combination, **client machine 10** determines, based on the null segment timeout value (“metadata”), a null segment timer (“timeout attribute”) associated with the RUDP (“second protocol”) connection. (EX1002 ¶211; *see also* Sections IX.A.1.e.)

Also the client determines the timeout value for a transfer state timer (“timeout attribute”) associated with the established RUDP connection (“second protocol connection”) based on the transfer state timeout value (“metadata”). (EX1007, 20:42–45, 23:60–61, 24:21–25; EX1002 ¶212; *supra* Sections IX.A.1.g.1.a, IX.A.1.g.2.a.) The client maintains a “transfer state timer” with a transfer state timeout value. (EX1007, 24:21–23; *see also id.*, 20:42–47; EX1002 ¶212.) Accordingly, in the *Wookey-Berg* combination, **client machine 10** determines, based on the transfer state timeout value (“metadata”), a transfer state timer (“timeout attribute”) associated with the RUDP (“second protocol”) connection. (EX1002 ¶213; *see also* Sections IX.A.1.e.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

A POSITA would have been motivated to implement the above null segment timeout value and/or transfer state timeout value in the combined *Wookey-Berg* system/method for the reasons discussed above. (See, e.g., *supra* Sections IX.A.1.e–g; EX1002 ¶214.)

* * *

Thus, the *Wookey-Berg* combination discloses and/or suggests the features of limitation 25.h as claimed. (EX1002 ¶215.)

2. Claim 28

a) **The apparatus of claim 25 wherein the code is communicated by permitting the client computer to fetch the code from the server computer utilizing the TCP connection and a hypertext transfer protocol (HTTP),²⁰**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶95, 216–218.) For example, as discussed for limitations 25.c–e, **client machine 10** and **remote machine 30** in the *Wookey-Berg* combination communicate over a HTTP based web browser application 280 utilizing the TCP connection to permit **client machine 10** to request and receive (“fetch”) code (e.g., the encoded URLs in HTML page 288 or HTML page 288 including the encoded URLs) from **remote machine 30** based on its request (e.g., request 282). (*Supra*

²⁰ See *supra* n.9.

Petition for *Inter Partes* Review
Patent No. 10,075,565

Sections IX.A.1.c–d (discussing that communication link 150 is established over a TCP connection), IX.A.1.e.1–2 (discussing that **client machine 10** uses HTTP at the application layer to communicate with **remote machine 30**.) Thus, the *Wookey-Berg* combination discloses and/or suggests this limitation for at least the same reasons discussed above for limitation 25.e.

- b) **and further wherein the code, when used by the client computer, causes the client computer to operate in accordance with the second protocol that is different from the TCP, by:**²¹

The *Wookey-Berg* combination discloses and/or suggests this limitation for at least the same reasons above for limitation 25.e. (*Supra* Section IX.A.1.e.1; EX1002 ¶219.)

- c) **detecting the idle time period based on the timeout attribute; and**

The *Wookey-Berg* combination discloses and/or suggests this limitation. (EX1002 ¶¶220–224.) As discussed for limitations 25.g and 25.h, **client machine 10** in the *Wookey-Berg* combination would configure both a null segment timer and a transfer state timer (each a “timeout attribute”) for detecting an idle time period. (*See supra* Sections IX.A.1.g–h.)

²¹ *See supra* n.9.

Petition for *Inter Partes* Review
Patent No. 10,075,565

First, as discussed above, in the combined *Wookey-Berg* system/process, **client machine 10** detects the idle time period defined by the null segment timeout value based on the expiration of the null segment timer (“timeout attribute”). (See *supra* Sections IX.A.1.g.1.a, IX.A.1.g.2.a; EX1007, 23:45–59; see also *id.*, 20:36–41; EX1002 ¶221.) Thus, “detecting the idle time period” is performed by detecting the null segment timer’s expiration. (EX1002 ¶221.)

Second, as discussed, in the combined *Wookey-Berg* system/process, **client machine 10** detects the idle time period defined by the transfer state timeout value based on the expiration of the transfer state timer (“timeout attribute”). (*Supra* Sections IX.A.1.g.1.b, IX.A.1.g.2.b; EX1007, 20:42–45, 24:15–25; EX1002 ¶222; *supra* Section IX.A.1.h.) Thus, “detecting the idle time period” is performed by detecting the transfer state timer’s expiration. (EX1002 ¶222.)

The above understanding is consistent with how the ’565 patent describes its “detecting.” (EX1001, 17:65–67 (“Detecting the first idle time period may include and/or otherwise may be based on detecting the timer expiration.”); EX1002 ¶¶223–224.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

- d) **deactivating the second protocol connection by communicating a particular packet between the client computer and the another server computer, in response to detecting the idle time period.**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶225–234.) As discussed, **client machine 10** (“client computer”) in the *Wookey-Berg* combination detects the idle time period based on detection of expiration of a null segment timer or the transfer state timer. (*Supra* Section IX.A.2.c.) As explained above and below, in response to **client machine 10** detecting the expiration of either the null segment timer or the transfer state timer, **client machine 10** deactivates the RUDP (“second protocol”) connection by communicating a RST segment or SYN segment (both disclosing a “particular packet”) between **client machine 10** and **remote machine 30’** (“another server computer”). (EX1002 ¶225.)

(1) Null Segment timer expiration

Berg explains that in response to detecting the null segment timer’s expiration (“in response to detecting the idle time period”), the client may deactivate the RUDP connection by communicating a RST segment (“a particular packet”) to the server (“packet between the client computer and the another server computer”) when the server fails to acknowledge the sent null segment. (EX1002 ¶¶226–229.) For example, when the client’s null segment timer expires, a null segment is sent to the

Petition for *Inter Partes* Review
Patent No. 10,075,565

server. (EX1007, 23:48–49.) The server must immediately acknowledge the null segment. (*Id.*, 22:27–30; *id.*, 23:49–51.) If the null segment is not acknowledged, the client may consider the connection broken. (*E.g.*, *Id.*, 22:27–30, 20:49–52; EX1002 ¶226.)

For instance, the Max Retrans value defines the “maximum number of times consecutive retransmissions” of a segment “will be attempted before the connection is considered broken.” (EX1007, 20:49–51.) When the negotiated number of retransmissions (Max Retrans) for the RUDP connection has a value of 1 (i.e., no retransmission of packets should be attempted), the client would determine the connection broken when no acknowledgment to the sent null segment is received. (EX1002 ¶227; EX1007, 20:49–52 (“An auto reset can be caused by the retransmission count exceeding its maximum”), 22:58–64 (“If this [retransmission] counter exceeds its maximum, the transmitter will perform a[n] auto reset on the connection and notify the Upper Layer Protocol (ULP) via the API.”).)

When the client determines that the connection is broken, the client will either reset or re-negotiate the RUDP connection based on the negotiated parameter, Max Auto Reset, for the RUDP connection. (EX1007, 21:18–22, 23:60–62; EX1002 ¶227.) When the Max Auto Reset is set to zero, then “the connection will be reset immediately” when an auto-reset condition occurs. (EX1007, 21:18–22.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

In the case where a reset/close (“deactivat[ion]”) of an RUDP connection is performed as a result of the server failing to acknowledge the client’s null segment, the client would communicate a RST segment (“particular packet”) to the server. (EX1007, 22:20; EX1002 ¶228; *supra* Section IX.A.1.g.) The RST segment is used to close or reset an RUDP connection. (EX1007, 22:20–24.) Accordingly, the “RST segment” would cause the deactivation of the connection to allow the establishment of a new connection. (*Id.*, 23:60–24:4; EX1002 ¶228.)

A POSITA would have thus understood that in the combined *Wookey-Berg* system/process, **client machine 10** would have performed a reset by resetting/closing (“deactivating”) the RUDP (“second protocol”) connection by communicating a RST segment (“particular packet”) between **client machine 10** (“client computer”) and **remote machine 30’** (“another server computer”), in response to detecting the idle time period set in the null segment timer. (EX1002 ¶229.)

(2) Transfer state timer expiration

Berg explains that in response to detecting the transfer state timer’s expiration (“in response to detecting the idle time period”), the client may deactivate the RUDP connection by communicating a SYN segment (“a particular packet”) to the server (“packet between the client computer and the another server computer”). (EX1002

Petition for *Inter Partes* Review
Patent No. 10,075,565

¶¶230–234.) For example, an auto-reset can be caused by the expiration of the transfer state timer. (EX1002 ¶230.) When the Max Auto Reset is set to a value greater than one, the client would be configured to perform several consecutive auto-resets equal to the Max Auto Reset value before a connection is reset. (EX1007, 21:15–16.) When the Max Auto Reset is set to at least a value of two in an RUDP connection, the expiration of the transfer state timer would have triggered a consecutive auto reset, at least one, before resetting the connection. (EX1002 ¶230.)

During an auto-reset, both the server and client will store the connection’s current state and attempt to transfer the saved state to another connection. (*Id.*, 23:62–67, 24:15–20.) In the case where an auto-reset (“deactivat[ion]”) of an RUDP connection is performed because of the expiration of the client’s transfer state timer, the client would have communicated to the server a SYN segment. (EX1007, 18:65–67 (“**A SYN segment is also used to perform an auto reset on a connection.**”); EX1002 ¶231.)

Thus the client, when configured to perform an auto-reset, would have performed an auto-reset (“deactivate”) on the RUDP (“second protocol”) connection by communicating a SYN segment (“particular packet”) to the server (“between the client computer and the another server computer”) in response to detecting the expiration of the transfer state timer (“idle time period”). (EX1002 ¶231.) A

Petition for *Inter Partes* Review
Patent No. 10,075,565

POSITA would have understood that such an auto-reset would cause the deactivation of the connection to allow the establishment of a new connection. (*Id.*; EX1007, 23:60–24:4, 18:65–67; *supra* Section IX.1.g.2.a (explaining that the auto-reset would cause the deactivation of the RUDP connection).)

A POSITA would have thus understood that in the combined *Wookey-Berg* system/process, **client machine 10** would have performed an auto-reset (“deactivating”) of the RUDP (“second protocol”) connection by communicating a SYN segment (“particular packet”) between **client machine 10** (“client computer”) and **remote machine 30’** (“another server computer”), in response to detecting the idle time period set in the transfer state timer. (EX1002 ¶232.)

* * *

Accordingly, based on the disclosures of *Berg* in context of *Wookey* (as explained above and for claim 25), a POSITA would have been motivated and found obvious (for reasons similar to those discussed above concerning the rationale for incorporating features disclosed/suggested by *Berg* with *Wookey*’s system/process) to configure the “code,” when used by **client machine 10**, that would result in the web browser application 280 to operate in accordance with the RUDP protocol to deactivate the RUDP (“second protocol”) connection by communicating a RST segment or a SYN segment (both disclosing a “particular packet”) between **client**

Petition for *Inter Partes* Review
Patent No. 10,075,565

machine 10 (“client computer”) and **remote machine 30’** (“another server computer”), in response to **client machine 10** detecting the expiration of either the null segment timer or the transfer state timer. (EX1002 ¶¶233–234; *supra* Section IX.A.1.e.)

X. DISCRETIONARY DENIAL IS NOT APPROPRIATE

The Board should not exercise its discretion to deny institution under 35 U.S.C. § 314(a). *NHK Spring Co., Ltd. v. Intri-Plex Techs, Inc.*, IPR2018-00752, Paper 8 (Sept. 12, 2018) does not apply here as no trial date has been scheduled in the related litigation. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 3 (Mar. 20, 2020) (precedential) (“*NHK* applies ... where the district court has set a trial date to occur earlier than the Board’s deadline to issue a final written decision in an instituted proceeding.”). The six-factor test addressed in *Fintiv* (“*Fintiv* factor(s)”) favors institution. *See id.*, 5–6.

The **first *Fintiv* factor** (stay) is neutral, if not weighing in favor of institution. While there is currently no stay pending IPR resolution before the Western District of Texas (“WDTX”) court, Google intends to pursue such a stay should institution be granted. The Board should not speculate as to how the court would rule on the requested stay based on actions taken in different cases. *Western Digital Corp. et al. v. Martin Kuster*, IPR2020-01391, Paper 10 at 8-9 (February 16, 2020) (finding this factor neutral despite Patent Owner assertions regarding WDTX Judge Albright’s past practices regarding stays). Moreover, Google filed a motion to transfer in the district court litigation. The Court has currently stayed the litigation pending resolution of Google’s transfer motion. (EX1030.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

The **second *Fintiv* factor** (proximity of trial dates) weighs strongly in favor of institution. *First*, the district court has not set a trial date, which “weighs significantly against exercising [] discretion to deny institution of the proceeding.” *Google LLC v. Uniloc 2017 LLC*, IPR2020-00441, Paper 13 at 35 (July 17, 2020); *Nested Bean, Inc. v. Big Beings USA PTY LTD*, IPR2020-01234, Paper 15 at 14 (Jan. 25, 2021) (same). In fact, while the parties submitted proposed schedules, the court granted Google’s motion to stay the case pending resolution of Google’s transfer motion and declined to set a case schedule.²² (EX1030; EX1024.) Accordingly, determining at this time whether the Board’s anticipated FWD date (e.g., Sept.–Oct. 2022) will occur after trial in the related litigation would be speculative.

Even if the parties’ most recent proposed schedule was considered as a baseline for comparison (which Google submits would be improper at this time), dates in that proposed schedule would need to be delayed because they were proposed prior to the court’s recent stay order and did not take the stay into consideration. (EX1030; EX1024, 5-9.) Accordingly, the proposed dates in light of

²² Disposition of Google’s transfer motion has taken priority over other activities. (EX1030.) *See also In re Apple Inc.*, 979 F.3d 1332, 1337 (Fed. Cir. 2020) (“disposing of a filed transfer motion ‘should unquestionably take top priority.’”).

Petition for *Inter Partes* Review
Patent No. 10,075,565

the court’s stay order demonstrates that trial will likely occur after August 2022.²³ (EX1024, 6, 9.) Thus, any projected trial date (*albeit* speculative) would be more likely after the expected due date of the Board’s FWD (e.g., around September–October 2022).

The Board has instituted trial even where an *actual* litigation trial date precedes the FWD deadline by two months or more. *Sand Revolution II, LLC v. Cont’l Intermodal Grp–Trucking LLC*, IPR2019-01393, Paper 24 at 8-10 (June 16, 2020) (“*Sand*”) (informative) (factor 2 favored institution despite WDTX trial preceding FWD deadline by five months); *Western Digital Corp.*, IPR2020-01391 at 9-10 (factor 2 found to be neutral despite WDTX trial predating FWD by three and a half months); *Apple Inc. v. Parus Holdings, Inc.*, IPR2020-00686, Paper 9 at 11–13, 22 (instituting trial where WDTX trial was two months before FWD deadline); *SMIC, Am. v. Innovation Foundry Techs., LLC*, IPR2020-00786, Paper

²³ Consistent with the court’s practice, the proposed order stages deadlines and trial date based on the proposed August 2021 *Markman* hearing and that the “actual trial date” may materially differ from the schedule and that the court would consider “reasonable amendments to the case schedule post-*Markman*.” (EX1024, 9 n.3; *id.*, 6.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

10, at 20–21 (Oct. 5, 2020) (same, where WDTX trial date preceded FWD by one month); *Fintiv*, IPR2020-00019, Paper 11 at 5, 9 (“an early trial date” is “non-dispositive” and simply means that “the decision whether to institute will likely implicate other factors”).

It is also unlikely that trial in WDTX will proceed without delay. WDTX has issued a suspension order every month for the past eleven months suspending almost all trials in the district from March 13, 2020 to at least March 31, 2021 due to the COVID-19 pandemic, creating a large backlog of trials. (*See* EX1025; *see also* EX1026, 2 (agreeing to stipulation to postpone trial date “to allow the COVID-19 situation to ameliorate”); EX1029 (showing J. Albright’s docket including 792 cases filed in 2020).) Thus, as the Board has recognized, civil trials in WDTX “may possibly slip ... [due to] months of backlogged trials, including many active criminal cases that would take precedence over civil trials.” *HP Inc. v. Slingshot Printing LLC*, IPR2020-01085, Paper 12 at 7 (Jan. 14, 2021); *Sand* at 8-10.

Finally, the potential for transfer based on Google’s transfer motion further reduces the likelihood that trial will precede the Board’s FWD. *Dish Network, L.L.C. v. Broadband iTV, Inc.*, IPR2020-01267, Paper 15 at 17-18 (Jan. 21, 2021) (explaining that the second *Fintiv* factor “is a proxy for the *likelihood* that the trial court will reach a decision on validity issues before the Board reaches a [FWD]”).

Petition for *Inter Partes* Review
Patent No. 10,075,565

Indeed, transfer of the WDTX case to the proper venue (the Northern District of California) will undoubtedly delay any trial beyond the anticipated FWD due date. *C.f. Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 8, at 7 n.7 (Mar. 27, 2020) (denying institution based in part on scheduled trial date that was later vacated upon transfer); *see Uniloc 2017 LLC v. Google LLC*, Case No. 18-CV-00502, Dkt No. 277 (E.D. Tex. June 19, 2020) (transfer order vacating unreached deadlines).

The **third *Fintiv* factor** (investment in parallel proceedings) also weighs in favor of institution. The case is in earliest stages, and as discussed, all deadlines other than those concerning Google’s transfer motion have been stayed. Thus, investment by the parties in invalidity and claim constructions issues will be delayed, especially if the case is transferred. And even if the transfer motion is denied, using the above-discussed proposed schedule as an exemplary baseline (which will require adjustments), any *Markman* hearing in the case is likely not to occur until August 2021 or later, and thus would occur near the expected due date of the Board’s institution decision. (EX1024, 6.) Discovery will not commence, and dispositive motions will not be due, until after the *Markman* hearing. Other case activity that has taken place or will eventually take place (outside claim construction) that does not relate to validity should “not weigh in [the Board’s] consideration of this issue.” *Western Digital Corp.*, IPR2020-01391 at 11. Such minimal investment is

Petition for *Inter Partes* Review
Patent No. 10,075,565

insufficient to support discretionary denial. *Juniper Networks, Inc. v. Huawei Digital Techs. (Cheng Du) Co., Ltd.*, IPR2020-01130, Paper 13 at 12-13 (Jan. 22, 2021) (factor 3 weighs against discretionary denial when claim construction order has been entered, but fact and expert discovery has not closed, and dispositive motions and briefings have not been submitted); *Dish* at 19-21 (similar); *HP* at 7 (similar). Additionally, Google’s diligence in filing this Petition just three months after receiving PO’s narrowed list of asserted claims²⁴ further weighs against discretionary denial. *Dish* at 20-21 (petitioner’s diligence in filing the petition weighed against discretionary denial); *Fintiv* at 11.

The **fourth *Fintiv* factor** (overlap) is neutral if not in favor of institution. There is, at this early stage of the district court litigation, no evidence of overlap with

²⁴ PO initially asserted 455 claims across eight patents (including claims 1–26 and 28–30 claims from the ’565 patent). (EX1017, 1.) The WDTX court ordered PO to substantially reduce the asserted claims to approximately 65 claims. On October 20, 2020, PO narrowed its asserted claims as ordered, but “reserve[d] the right to either narrow the claims further, **substitute claims**, or **add a reasonable number of claims**.” (EX1027.) On December 4, 2020, PO informed Petitioner that it will not modify its narrowed list of asserted claims “**at this time**.” (EX1028.)

Petition for *Inter Partes* Review
Patent No. 10,075,565

the grounds here and any invalidity positions to be pursued in the district court. Indeed, preliminary invalidity contentions have not yet been served. And, as discussed above, the district court must first resolve the pending transfer motion before addressing substantive issues, including invalidity. Given the stay of the litigation, it is likely expert discovery relating to invalidity will not be completed until at the earliest mid-2022 and dispositive motions on those issues would be due after that. Accordingly, substantive invalidity issues would not be addressed in the litigation until well past the institution decision and likely very near the time of the Board's expected FWD. Moreover, Petitioner may rely on invalidity grounds not asserted here in the district court litigation, including system art and prior art references and combinations not raised herein. Therefore, any potential overlap between this proceeding and the district court is minimal and speculative at this time.

Other circumstances (***Fintiv* factor six**) also favor institution. The Petition presents strong grounds demonstrating the unpatentability of the challenged claims, which weighs against discretionary denial. (*See supra* Section IX.) *Western Digital Corp.*, IPR2020-01391 at 14-15. Indeed, the Board has already instituted the Google -845 IPR and the Unified -742 IPR challenging claims of the related '995 patent. (*See supra* Section II.) The Unified -742 IPR is based in part on *Berg*, which is being applied here. Moreover, this petition is the only challenge to the '565 patent

Petition for *Inter Partes* Review
Patent No. 10,075,565

before the Board, which is a “crucial fact” favoring institution. *Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 10 at 6 (May 12, 2020).²⁵

While ***Fintiv* factor 5** (same parties) may weigh slightly in favor of denial, the remaining factors are at least neutral, if not strongly favoring institution. Further, even if the Board finds that **factor 1** (or even **factor 2**) somehow tilts in favor of denial, the investment of resources, lack of evidence of overlap, and/or strength of Petitioner’s unpatentability positions (**factors 3, 4, 6**) outweigh these other factors. Accordingly, based on a “holistic view” of whether integrity of the system and efficiency is best served, institution here is proper. *Samsung Elecs. Co. Ltd. v. Dynamics Inc.*, IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

²⁵ Discretionary denial under *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper No. 19 is not applicable here given the ’565 patent is not at issue in any other proceeding pending before the Board.

Petition for *Inter Partes* Review
Patent No. 10,075,565

XI. CONCLUSION

For the reasons given above, Petitioner requests institution of IPR for the challenged claims based on each of the specified grounds.

Respectfully submitted,

Dated: March 15, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,075,565

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,075,565 contains, as measured by the word-processing system used to prepare this paper, 13,968 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: March 15, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,075,565

CERTIFICATE OF SERVICE

I hereby certify that on March 15, 2021, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,075,565 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on PAIR:

Patrick Caldwell (Reg. No. 44580)
G. Gordon (Reg. No. 64517)
P.O. Box 59655
Dept. SVIPGP
Dallas, TX 75229

A courtesy copy was also sent via electronic mail to the Patent Owner's litigation counsel at the following addresses:

TD-PTAB@devlinlawfirm.com
ddahlgren@devlinlawfirm.com
dlflitparas@devlinlawfirm.com
1526 Gilpin Ave.
Wilmington, DE 19806

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)

EXHIBIT 6

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

JENAM TECH, LLC,
Patent Owner

Patent No. 10,375,215

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,375,215**

Petition for *Inter Partes* Review
Patent No. 10,375,215

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	1
III.	PAYMENT OF FEES	2
IV.	GROUND FOR STANDING.....	2
V.	PRECISE RELIEF REQUESTED	3
	A. Claims for Which Review Is Requested	3
	B. Statutory Grounds of Challenge.....	3
VI.	LEVEL OF ORDINARY SKILL	5
VII.	OVERVIEW OF THE '215 PATENT	5
VIII.	CLAIM CONSTRUCTION	9
IX.	DETAILED EXPLANATION OF GROUNDS.....	12
	A. Ground 1: The Combination of <i>Wookey</i> and <i>Eggert</i> Render Obvious Claim 1	12
	1. Claim 1	12
	B. Ground 2: The Combination of <i>Wookey</i> , <i>Eggert</i> , and <i>Abdolbaghian</i> Render Obvious Claims 4, 8, and 9.....	53
	1. Claim 4.....	53
	2. Claim 8.....	65
	3. Claim 9.....	66
X.	EGGERT IS A PRINTED PUBLICATION	68
XI.	DISCRETIONARY DENIAL IS NOT APPROPRIATE	69
	A. <i>Eggert</i> is Not Cumulative.....	69

Petition for *Inter Partes* Review
Patent No. 10,375,215

B.	The Related Litigation Provides No Basis For Discretionary Denial	71
XII.	CONCLUSION.....	80

Petition for *Inter Partes* Review
Patent No. 10,375,215

LIST OF EXHIBITS

EX1001	U.S. Patent No. 10,375,215
EX1002	Declaration of Bill Lin, Ph.D.
EX1003	Curriculum Vitae of Bill Lin, Ph.D.
EX1004	Prosecution History of U.S. Patent No. 10,375,215
EX1005	U.S. Pre-Grant Publication No. 2007/0171921 to Wookey <i>et al.</i> (“Wookey”)
EX1006	L. Eggert, TCP Abort Timeout Option, draft-eggert-tcm- tcp-abort-timeout-option-00, Network Working Group, Internet-Draft (April 14, 2004) (“Eggert”)
EX1007	U.S. Patent No. 6,981,048 to Abdolbaghian (“Abdolbaghian”)
EX1008	DARPA RFC 793 TRANSMISSION CONTROL PROTOCOL
EX1009	RFC 1122 “Requirements for Internet Hosts -- Communication Layers”
EX1010	U.S. Patent No. 9,923,995
EX1011	RFC 1001
EX1012	U.S. Patent No. 6,212,175 to Harsch
EX1013	U.S. Patent No. 6,665,727 to Hayden
EX1014	U.S. Patent No. 7,636,805 to Rosenberg
EX1015	RFC 5440
EX1016	IETF RFC 5482

Petition for *Inter Partes* Review
Patent No. 10,375,215

EX1017	<i>Jenam Tech, LLC's First set of Infringement Contentions</i> regarding U.S. Patent No. 10,375,215 (August 21, 2020)
EX10181	<i>Jenam Tech, LLC v. LG Electronics, Inc. et al</i> , Case No. 4:19-cv-00249, ECF No. 1 (E.D. Tex. Apr. 3, 2019)
EX1019	Declaration of Alexa Morris for Eggert
EX1020	U.S. Pre-Grant Publication No. 2004/0093376 to De Boor <i>et al.</i>
EX1021	U.S. Patent 7,535,913 to Minami <i>et al.</i>
EX1022	U.S. Pre-Grant Publication No. 2005/0204013 to Raghunath <i>et al.</i>
EX1023	U.S. Pre-Grant Publication No. 2007/0005804 to Rideout
EX1024	U.S. Pre-Grant Publication No. 2004/0098748 to Bo <i>et al.</i>
EX1025	IETF RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1
EX1026	U.S. Patent No. 8,259,716 to Diab
EX1027	Bova et al., RELIABLE UDP PROTOCOL <draft-ietf-sigtran-reliable-udp-00.txt> 25 February 1999
EX1028	U.S. Patent 6,674,713 to Berg et al.
EX1029	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Joint Submission of Proposed Amended Scheduling Order, ECF No. 57 (W.D. Tex. Mar. 4, 2021)
EX1030	Thirteenth Supplemental Order Regarding Court Operations Under the Exigent Circumstances Created By the COVID-19 Pandemic, The United States District Court for the Western District of Texas (February 2, 2021)
EX1031	<i>Digital Retail Apps, Inc. v. H-E-B, LP</i> , Case No. 6:19-cv-00167, Joint Stipulation and Order Postponing Trial, ECF No. 182 (W.D. Tex. Jan. 13, 2021)
EX1032	<i>Jenam Tech, LLC's Preliminary Narrowing of Claims</i> (October 20, 2020)

Petition for *Inter Partes* Review
Patent No. 10,375,215

EX1033	<i>Jenam Tech, LLC's</i> Correspondence Regarding Narrowing of Claims (December 4, 2020)
EX1034	Judge Alan D. Albright's Case Statistics By Year (Retrieved from DocketNavigator on March 9, 2021)
EX1035	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Order Granting Motion to Stay Case, ECF No. 58 (W.D. Tex. Mar. 10, 2021)

Petition for *Inter Partes* Review
Patent No. 10,375,215

I. INTRODUCTION

Google LLC (“Petitioner”) requests *inter partes* review (“IPR”) of claims 1, 4, 8, and 9 (“the challenged claims”) of U.S. Patent No. 10,375,215 (“the ’215 patent”) (EX1001) assigned to Jenam Tech, LLC (“Patent Owner” or “PO”). For the reasons below, the challenged claims should be found unpatentable and canceled.

II. MANDATORY NOTICES

Real Parties-in-Interest: Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies the following as the real parties-in-interest: Google LLC.¹

Related Matters: The ’215 patent is asserted in the following civil action: *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.). The ’215 patent was previously asserted in the following civil action: *Jenam Tech, LLC v. Samsung Electronics Co., Ltd.*, Case No. 4:20-cv-00279 (E.D. Tex.) (voluntarily dismissed).

The ’215 patent claims priority to U.S. Patent Application No. 15/694,802, filed on September 3, 2017, and issued as U.S. Patent No. 9,923,995 (“the ’995

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

Petition for *Inter Partes* Review
Patent No. 10,375,215

patent”). The ’995 patent is subject to the following instituted IPRs: *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845 (“Google -845 IPR”); and *Unified Patents, LLC. v. Jenam Tech, LLC*, IPR2020-00742 (“Unified -742 IPR”).

Petitioner is also concurrently filing IPR petitions challenging U.S. Patent Nos. 10,075,564, and 10,075,565 which are also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.).

Counsel and Service Information: Lead counsel: Naveen Modi (Reg. No. 46,224). Backup counsel: (1) Joseph E. Palys (Reg. No. 46,508), (2) Quadeer A. Ahmed (Reg. No. 60,835), (3) Jason Heidemann (Reg. No. 77,880).

Service information is Paul Hastings LLP, 2050 M Street NW, Washington, DC 20036, Tel.: 202.551.1700, Fax: 202.551.1705, email: PH-Google-Jenam-IPR@paulhastings.com. Petitioner consents to electronic service.

III. PAYMENT OF FEES

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-2613.

IV. GROUNDS FOR STANDING

Petitioner certifies that the ’215 patent is available for review and Petitioner is not barred or estopped from requesting review on the grounds identified herein.

V. PRECISE RELIEF REQUESTED

A. Claims for Which Review Is Requested

Petitioner respectfully requests review of claims 1, 4, 8, and 9 (“the challenged claims”) of the ’215 patent and cancellation of those claims as unpatentable.

B. Statutory Grounds of Challenge

The challenged claims should be canceled as unpatentable in view of the following grounds:

Ground 1: Claim 1 is unpatentable under pre-AIA 35 U.S.C. § 103(a) over U.S. Publication No. 2007/0171921 to Wookey *et al.* (“*Wookey*”) (EX1005) in view of Eggert, TCP Abort Timeout Option (April 14, 2004) (“*Eggert*”) (EX1006); and

Ground 2: Claims 4, 8, and 9 are unpatentable under pre-AIA 35 U.S.C. § 103(a) over *Wookey* in view of *Eggert* and U.S. Patent No. 6,981,048 to Abdolbaghian (“*Abdolbaghian*”) (EX1007).²

The ’215 patent issued from U.S. Patent Application No. 16/040,522 (“the ’522 application”), filed on July 19, 2018. (EX1001, Cover.) The ’522 application is a continuation of U.S. Patent Application No. 15/915,047, filed on March 7, 2018,

² Other references discussed herein are provided to show the state of the art at the time of the alleged invention of the ’215 patent.

Petition for *Inter Partes* Review
Patent No. 10,375,215

which is a continuation of application No. 15/694,802 (“the ’802 application”), filed on September 3, 2017. (*Id.*) The ’802 application is a continuation-in-part (CIP) of application No. 14/667,642, filed on March 24, 2015, which is a CIP of application No. 13/477,402, filed on May 22, 2012, which is a continuation of application No. 12/714, 454 (“the ’454 application”), filed on February 27, 2010. (*Id.*, *see also id.*, 1:8–34.)

For purposes of this proceeding only, Petitioner assumes, without conceding, the earliest effective filing date of the ’215 patent is February 27, 2010, (filing date of the ’454 application).

Wookey was published on July 26, 2007, from an application filed on November 14, 2006. (EX1005, Cover.) *Abdolbaghian* issued December 27, 2005, from an application filed on November 22, 2000. (EX1007, Cover.) Therefore, both *Wookey* and *Abdolbaghian* qualify as prior art at least under 35 U.S.C. § 102(b) (pre-AIA).

Eggert, which is an Internet Engineering Task Force (IETF) Internet-Draft (or “ID”) working document, was published on April 14, 2004. (EX1006, 1; *see also infra* Section X.) As confirmed by the declaration of Alexa Morris, Managing Director of IETF, *Eggert* was published, disseminated, and reasonably available to

Petition for *Inter Partes* Review
Patent No. 10,375,215

the public by April 15, 2004. (EX1019, ¶¶9–10; *see also infra* Section X.) Thus, *Eggert* qualifies as prior art at least under 35 U.S.C. § 102(b) (pre-AIA).

The above references were not considered during prosecution of the '215 patent (EX1001, Cover (“References Cited”); *see also generally* EX1004) and are not the same or substantially similar to any art previously presented to the Office. (*See infra* Section XI.A.)

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art at the time of the alleged invention of the '215 patent (“POSITA”) would have an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking. (EX1002 ¶¶16–18.)³ More education can supplement practical experience and vice versa. (*Id.*)

VII. OVERVIEW OF THE '215 PATENT

The '215 patent is directed to networking, and in particular, to the sharing of information for detecting an idle TCP connection. (EX1001, 2:32–34, 8:37–58.) Figure 7 illustrates such a process.

³ Petitioner submits the declaration of Dr. Bill Lin (EX1002), an expert in the field of the '215 patent. (EX1002 ¶¶3–15; EX1003.)

Petition for *Inter Partes* Review
 Patent No. 10,375,215

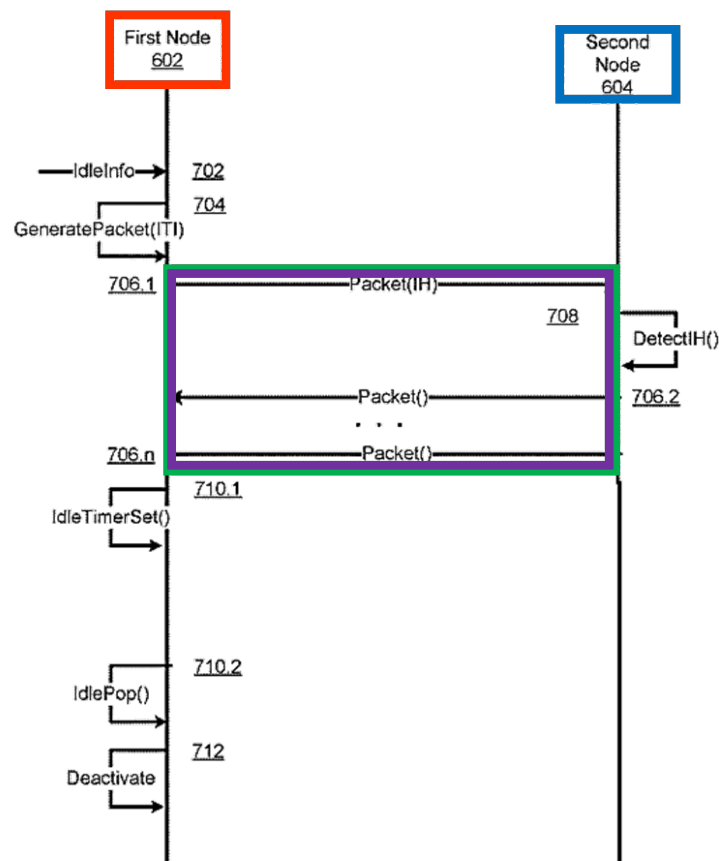


Fig. 7

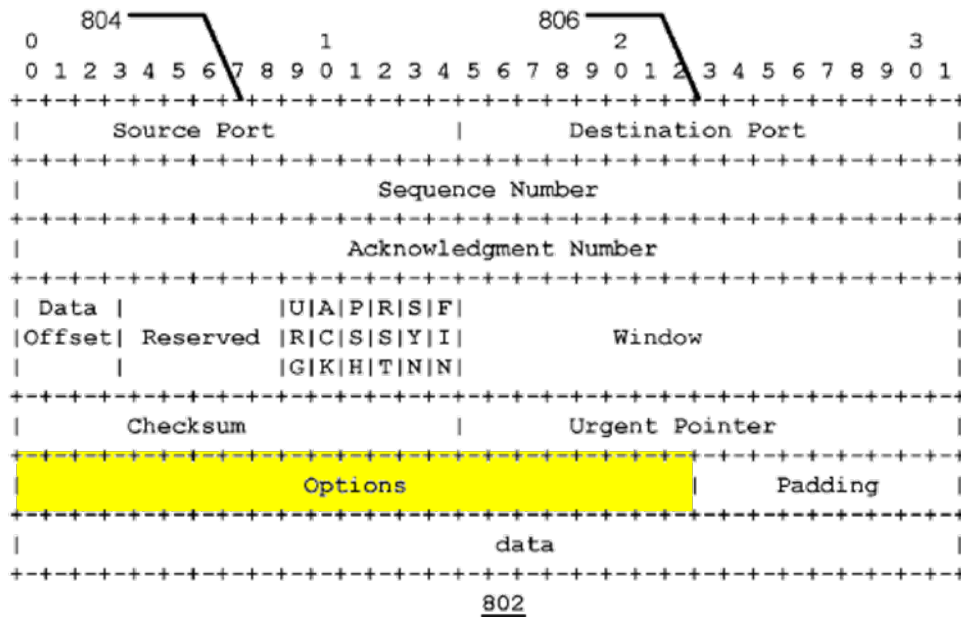
(*Id.*, FIG. 7 (annotated), 6:31–35; EX1002 ¶64.)

First, the **first node 602** receives a message 702 that identifies idle information representing a duration for an idle time period (“ITP”). (EX1001, 11:62–64.) The idle information “may include and/or identify a duration of time for detecting an idle time period.” (*Id.*, 12:42–47.) The “duration may be specified

Petition for *Inter Partes* Review
Patent No. 10,375,215

according to various measures of time[,] including seconds, minutes, hours, and/or days.” (*Id.*; EX1002 ¶65.)

Next, the **first node** “generat[es] a TCP packet including an ITP header based on received idle information.” (EX1001, 16:10–14.) The **TCP options field** of a TCP packet may store this ITP header. (*Id.*, 15:31–51.) Figure 8 below, “adapted from RFC 793,” illustrates this configuration. (*Id.*, FIG. 8, 6:35–37, 15:31–51.)



Petition for *Inter Partes* Review
 Patent No. 10,375,215

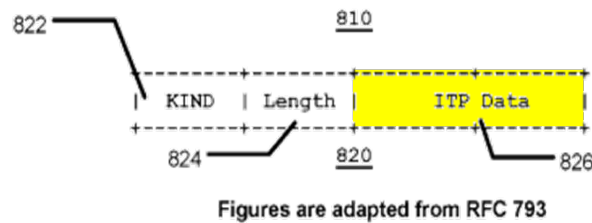


Fig. 8

(*Id.*, FIG. 8 (cropped and annotated); EX1002 ¶67.)

The ITP header is exchanged during the three-way handshake (EX1001, 14:56–67) or “in a TCP packet in structures and locations other than those specified for TCP options in RFC 793” (*id.*, 15:61–64). For example, as shown in Figure 7 above, the **first node** transmits a message 706.1, i.e., a TCP packet including an ITP header (IH) that contains ITP information, to the **second node 604**. (*Id.*, 16:35–36.) Message 708 exemplifies the **second node**’s detection of the IH in the received TCP packet. (*Id.*, 21:58–63; EX1002 ¶68.)

The ’215 patent explains that “a TCP keep-alive option, a TCP user timeout, a retransmission timeout, an acknowledgment timeout, and/or another timeout associated with a TCP connection may be modified based on the first idle information.” (EX1001, 13:54–57; *see also id.*, 22:23–28.) And “the modifying may be based on the content of the ITP header.” (*Id.*, 22:28–30; EX1002 ¶69.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

The challenged claims recite limitations relating to features discussed above. However, all the limitations in the challenged claims were known in the prior art and obvious. (*See* Section IX; EX1002 ¶70; *see also id.* ¶¶19–63 (discussing technology background), citing, e.g., EXS. 1020–1028.)

VIII. CLAIM CONSTRUCTION

Under the applicable standard in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), claim terms are typically given their ordinary and customary meanings as understood by a POSITA at the time of the invention based on the claim language, specification, and the prosecution history of record. *Phillips*, 415 F.3d at 1313; *see also id.* at 1312–16. The Board, however, only construes the claims when necessary to resolve the controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper No. 11 at 16 (Aug. 14, 2015) (citing *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)). Petitioner believes that no express constructions of any claim terms are necessary to assess whether the prior art reads on the challenged claims.⁴ (EX1002 ¶71.)

⁴ Petitioner reserves all rights to raise claim construction and other arguments, including challenges under 35 U.S.C. §§ 101 or 112, in district court as relevant to those proceedings. *See, e.g., Target Corp. v. Proxicom Wireless, LLC*, IPR2020-

Petition for *Inter Partes* Review
Patent No. 10,375,215

Challenged claim 1 recites the terms “TCP-variant packet” and “TCP-variant connection” (EX1001, 24:21–31), which are also recited in claims of the related ’995 patent at-issue in the Unified -742 IPR and Google -845 IPR (*see supra* Section II). The ’215 patent and the ’995 patent share a common specification. (*Compare* EX1001 *with* EX1010 (’995 patent).)

The ’215 patent discloses that the “TCP-variant” protocol varies slightly from TCP by adding an idle timeout option using the format for new TCP options in a TCP header. (EX1001, 15:31–60, 16:10–14, FIG. 8; *see also supra* Section VII.)

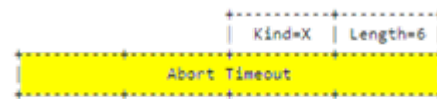
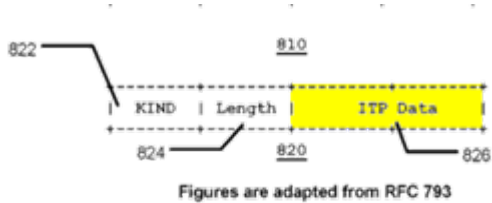
Like the Board noted for the ’995 patent, the ’215 patent’s specification “provides little guidance as to [the] meaning [of ‘TCP-variant connection’].” Google -845 IPR, Paper 16 (Institution Decision) at 17 (October 8, 2020); Unified -742 IPR, Paper 11 (Institution Decision) at 5 (October 8, 2020). (*See generally* EX1001.)⁵

00904, Paper 11 at 11–13 (November 10, 2020). A comparison of the claims to any accused products in litigation may raise controversies that are not presented here given the similarities between the references and the patent.

⁵ While the ’215 patent generically refers to “variant of the current TCP” (EX1001, 16:66–67), the term “TCP-variant” is only found in the claims.

Petition for *Inter Partes* Review
 Patent No. 10,375,215

Eggert's timeout option is a similar variation on TCP as disclosed in the '215 patent and thus discloses the term "TCP-variant." (*See, e.g., infra* Sections IX.A.1.d–f.) As shown below, *Eggert*'s timeout option (right) uses the same TCP structure as the ITP field illustrated in Figure 8 of the '215 patent (left), including a data portion (YELLOW) containing the timeout value.



EX1001, FIG. 8 (annotated)

EX1006, FIG. 1 (annotated)

(EX1002 ¶¶91, 147; *infra* Section IX.A.1.d.) Moreover, the Board preliminarily found that *Eggert* discloses these TCP-variant limitations in the Google -845 IPR. *See, e.g.,* Google -845 IPR, Paper 16 at 20–23.

Thus, given that the '215 patent offers no descriptions that would require a special meaning of these terms and that the prior art discloses these features under any reasonable interpretation, the Board need not construe the TCP-variant terms. Instead, as Petitioner does, the Board should apply the plain meaning of the terms "TCP-variant packet" and "TCP-variant connection" like the Board has applied in the Unified -742 IPR and Google -845 IPR. (*See infra* Sections IX.A.1.d–f.)

IX. DETAILED EXPLANATION OF GROUNDS**A. Ground 1: The Combination of *Wookey* and *Eggert* Render Obvious Claim 1****1. Claim 1****a) “A computer-implemented method, comprising:”**

To the extent that the preamble of claim 1 is limiting, *Wookey* discloses the limitations therein. (EX1002 ¶¶100–106.)

Wookey discloses a computer-implemented method, as claimed. For example, *Wookey* teaches “a *method* for making a hypermedium page interactive.” (EX1005, Abstract; *see also id.* ¶[0002].)⁶ The disclosed method is implemented using a client machine and one or more remote machines. (*Id.*, Abstract; EX1002 ¶102; *see also id.* ¶¶72–81 (*Wookey* overview).)

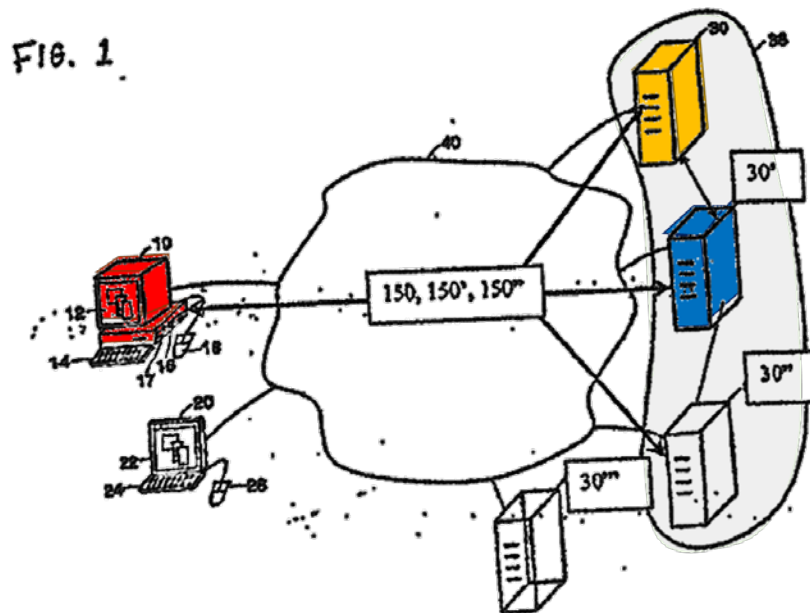
Figures 1, 2C, and 3D below show an exemplary environment in which a **client machine 10**, **remote machine 30**, and **remote machine 30'**, amongst other components, are employed to carry out the method.⁷ (*See also* EX1005 ¶¶[0016],

⁶ Emphasis is added unless otherwise stated.

⁷ *Wookey*'s disclosures (including its figures) include certain typographical errors when referencing its client machines (referring to them interchangeably as 10, 10' or 20). (EX1002 ¶113; EX1005 ¶¶[0135]–[0136], [0152]–[0154], [0192]–[0196],

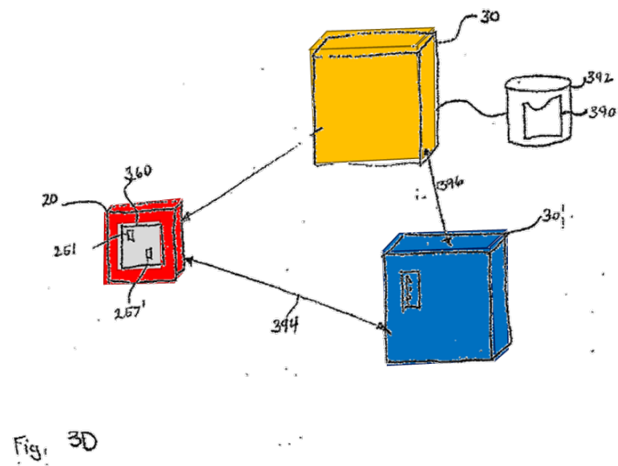
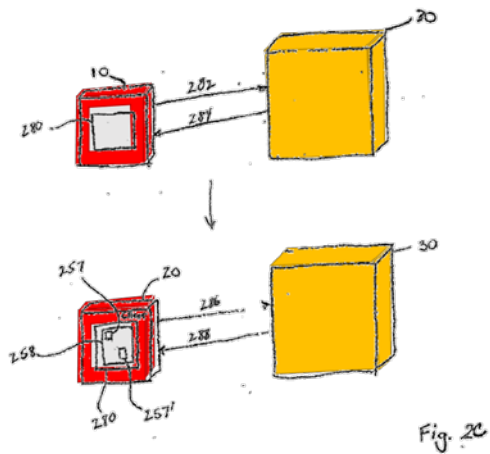
Petition for *Inter Partes* Review
 Patent No. 10,375,215

[0019], [0020], [0024], [0026], [0192]–[0196], [0207], [0213].) *Wookey* discloses that the client and remote machines are “typical computers” (*id.* ¶[0021], FIGS. 1A–1B) including personal computers and wireless devices (*id.* ¶[0171]) such as mobile telephone or other portable telecommunication devices (*id.* ¶¶[0172]–[0173]).



[0207]–[0216], FIGS. 1, 2C, 3D.) For example, the bottom figure in Fig. 2C erroneously shows client machine “20,” but the corresponding disclosure for both figures in Fig. 2C describes client machine “10.” (EX1005 ¶¶[0192]–[0196].) Moreover, in context of *Wookey*’s disclosures, a POSITA would have understood that any teachings regarding one client machine 10/10’/20 in *Wookey* apply equally to the other client machines. (EX1002 ¶113.)

Petition for *Inter Partes* Review
Patent No. 10,375,215



(EX1005, FIGS. 1 (annotated), 2C (annotated), 3D (annotated); EX1002 ¶¶103–105.)

Thus, *Wookey* discloses “[a] computer-implemented method,” as claimed.
(*See also infra* Sections IX.A.1.b–g; EX1002 ¶106.)

- b) **“at a server node: causing data to be sent to a client node; and”**

Wookey discloses this limitation. (EX1002 ¶¶107–116.) As explained below, *Wookey* discloses that **remote machine 30** (“server node”) causes an HTML page, which includes data, to be sent to a **client machine 10**. (*Id.*)

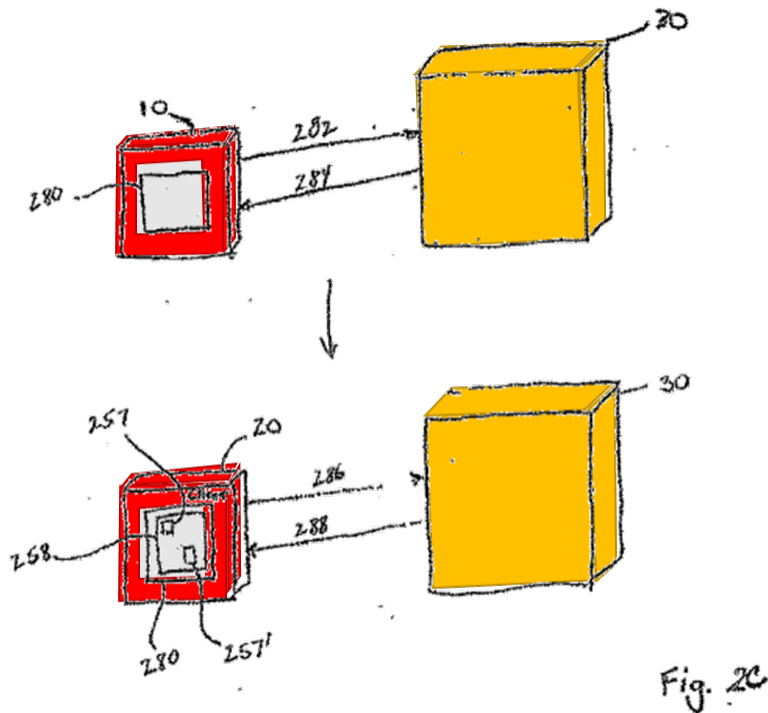
Remote machine 30 is a server node because *Wookey*’s remote machines (e.g., remote machine 30) “are provided as computers or *computer servers*.” (EX1005 ¶[0160]; *see also id.* ¶¶[0021], [0136] (**remote machine 30** is contained within a *server* farm 38 with other remote machines), [0150] (“Remote machines 30 may be servers, file servers, application servers, ... gateway servers, virtualization servers, deployment servers.”), FIGS. 1A–1B.) *Wookey* discloses that “remote machine 30 function[s] as a web server” to “receive[] communications from the client machine 10.” (*Id.* ¶[0178]; *see also id.* ¶[0136]; EX1002 ¶¶109–110.)⁸

Wookey discloses that **remote machine 30** causes data to be sent to **client machine 10**. For example, with exemplary reference to Figure 2C, **remote machine**

⁸ *Wookey* uses the term “machine” to describe what the ’215 patent refers to as a “node.” (EX1005 ¶[0153] (“The client machines 10 may also be referred to as endpoints, client nodes, clients, or local machines.”), [0301] (“any remote machine 30 (which may be referred to as a node)”); EX1002 ¶110 n.3.)

Petition for *Inter Partes* Review
 Patent No. 10,375,215

30 is configured to transmit data in the form of an HTML page to **client machine 10**, using a web browser application. (EX1005 ¶¶[0192]–[0193], [0207] (“remote machine 30 as described above in connection with FIG. 2C” “in this example acts as a web server”); *see also id.* ¶[0024].)



(*Id.*, FIG. 2C (annotated); EX1002 ¶¶111–114.)

Client machine 10, as illustrated in Figure 2C, executes a web browser application 280 (EX1005 ¶[0192]) that “transmits a request 282 to access a Uniform Resource Locator (URL) address corresponding to an HTML page residing on

Petition for *Inter Partes* Review
Patent No. 10,375,215

remote machine [30]” (*id.* ¶[0193]).⁹ Following this request from **client machine 10**, **remote machine 30** returns an HTML page (e.g., page 284) to **client machine 10**. (*Id.* ¶[0193]; EX1002 ¶¶112, 114.)

The HTML page returned may be “an authentication page [284] that seeks to identify the client machine 10 or the user of the client machine 10.” (EX1005 ¶[0193].) HTML page 284 includes data such as tags and associated information defining the layout of a webpage and what is to be presented on the webpage. (EX1002 ¶115; *see also* EX1005 ¶¶[0196], [0207]–[0214] (discussing an exemplary webpage that **remote machine 30** transmits to **client machine 10**), [0754]–[0755] (same).)

⁹ *Wookey*’s reference to “remote machine” as “remote machine 10” in the first sentence of paragraph [0193] is a typographical error. (EX1002 ¶112.) In the same paragraph, *Wookey* identifies the same remote machine as “the remote machine 30,” consistent with Figure 2C. (*Id.*; EX1005 ¶¶[0193], [0195]–[0196], FIGS. 2C, 3D.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

The understanding that an HTML page including data meets this limitation is consistent with PO’s interpretation of this limitation. (EX1017, 2 (alleging “at a server node: causing data (e.g., in HTML pages, etc.) to be sent to a client node.”).)¹⁰

Accordingly, *Wookey* discloses limitation 1.b. (EX1002 ¶116.)

- c) **“providing access, to the client node, to code that, in response to being used by the client node, causes the client node to:”**¹¹

Wookey discloses this limitation. (EX1002 ¶¶117–126.) As discussed below, *Wookey* discloses that **remote machine 30** (“server node”) provides access, to **client machine 10**, to encoded Uniform Resource Locators (URLs) (“code”) associated

¹⁰ Petitioner’s references to PO’s infringement allegations are not indicative of any concession to PO’s allegations that any accused instrumentality infringes any claim of the challenged patent.

¹¹ The ’215 patent specification does not provide any disclosure describing the claimed “code” features in the claim. (EX1002, ¶117 n.4.) Therefore, while Petitioner addresses such features with respect to the asserted prior art, Petitioner does not concede that the ’215 patent provides adequate disclosures of such claimed features.

Petition for *Inter Partes* Review
Patent No. 10,375,215

with icons on an HTML page 288 (where the HTML page 288 is also “code”)¹² that, in response to being used by **client machine 10**, causes **client machine 10** to initiate a process for establishing a second connection with another **remote machine 30**’ as discussed below with respect to limitations 1.d–g. (*Id.*; *infra* Sections IX.A.1.d–g.)

For example, after identifying **client machine 10**, **remote machine 30** may “transmit[] to the client machine 10 an HTML page 288 that includes a Resource Neighborhood window 258 in which appears graphical icons 257, 257’ representing resources to which the client machine 10 has access.” (EX1005 ¶[0196].) The Resource Neighborhood window 258 displays graphical icons 257, 257’ representing resources to which **client machine 10** has access. (*Id.* ¶[0196]; *see also id.* ¶¶[0215]–[0216].) Thus, **remote machine 30** provides access, to **client machine 10**, to encoded Uniform Resource Locators (URLs) (“code”) associated with icons on an HTML page 288 (where the HTML page 288 is also “code”). (EX1002 ¶¶119–120.)

¹² Petitioner’s analysis regarding the “code” limitation equally applies whether the “code” is the encoded URLs in the HTML page or the HTML page itself (which includes the encoded URLs). (EX1002 ¶120 n.5.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

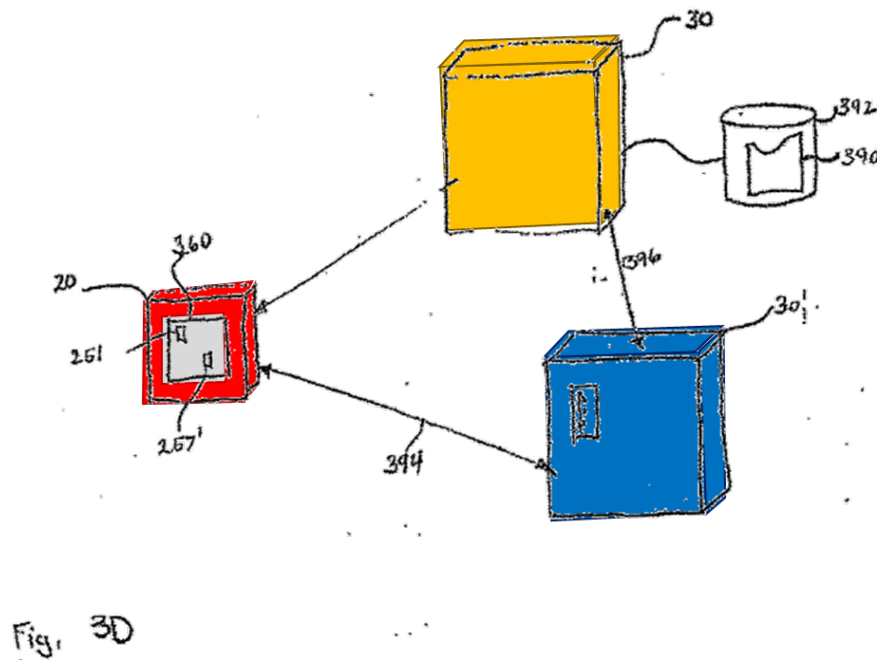
The HTML page 288 includes encoded URLs associated with icons (e.g., 257, 257') on the HTML page 288. These encoded URLs are the claimed “code that, in response to being used by the client node, causes the client node to” initiate a process for establishing a second connection with **remote machine 30'** and perform the steps set forth in limitations 1.d–f. (EX1002 ¶121; *infra* Sections IX.A.1.d–f.)

In particular, “[e]ach icon 257, 257' *is associated with* an encoded URL that specifies: the location of the resource ...; *a launch command* associated with the resource; and *a template identifying how the results of accessing the resource should be displayed.*” (EX1005 ¶[0216].) *Wookey* also discloses that each encoded URL “*contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (*Id.*) Thus, when a user *clicks* an icon 257, 257' corresponding to a resource from the displayed HTML page 288, **client machine 10** uses the encoded URL associated with the clicked icon to initiate the process for establishing a second connection (e.g., connection 394 in Figure 3D below) with a second **remote machine 30'**. (*Id.*; *see also id.* ¶[0026]; EX1002 ¶121.)

Accordingly, the encoded URLs are code “that, in response to being used by the client node, causes the client node to” perform certain steps because such URLs

Petition for *Inter Partes* Review
 Patent No. 10,375,215

are instructions that, when invoked, command **client machine 10** to access resources from one or more locations (e.g., one or more servers). (EX1002 ¶122.)



(EX1005, FIG. 3D (annotated); *see also id.* ¶[0026]; EX1002 ¶121.)

The above understanding is consistent with how the '215 patent describes its “code.” (EX1001, cl. 36 (29:25) (reciting that “the code includes one or more links”).) *Wookey*’s encoded URLs associated with the icons on the HTML page 288 similarly include a link (e.g., specifying the location of the remote resource and launch command to connect to the resource). (EX1002 ¶123.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

Moreover, HTML page 288 (including the encoded URLs) also discloses the claimed “code” because an HTML page was known to comprise HTML code used to define, e.g., what is displayed on the screen of the client node. (*Id.* ¶¶124 (citing EX1020 ¶¶[0022], [0024], [0026], [0079]).) The HTML code used to display information relating to page 288 would include the icons associated with the URLs that the client node uses to access a desired resource, as further discussed below for limitation 1.d. Accordingly, HTML page 288 includes code. (EX1002 ¶¶124–125.) This understanding (e.g., HTML page includes code) is consistent with PO’s interpretation of this limitation in its infringement positions for the ’215 patent. (EX1017, 2 (“code (e.g., also in the HTML pages, etc.)”).)¹³

Accordingly, *Wookey* discloses limitation 1.c. (EX1002 ¶126.)

d) **“receive, by the client node, a transmission control protocol (TCP)-variant packet,”**

The combination of *Wookey* and *Eggert* discloses and/or suggests this limitation. (EX1002 ¶¶127–171.) As explained above, when **client machine 10** uses the code, it results in **client machine 10** operating to initiate a process for establishing a second connection with another **remote machine 30**. (*Supra* Section IX.A.1.c.) And as discussed below, the process for establishing the second

¹³ See *supra* n.10.

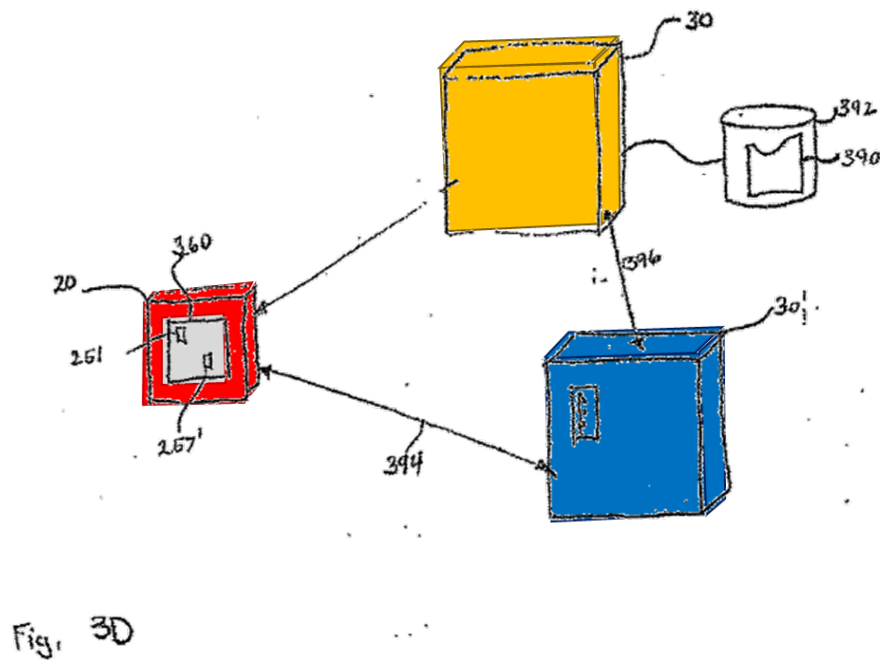
Petition for *Inter Partes* Review
Patent No. 10,375,215

connection includes receiving a TCP-variant packet, as claimed, in the *Wookey-Eggert* combination. (*See also infra* Sections IX.A.1.e–g (discussing additional steps in the process for establishing the connection).)

For example, regarding the connection between **client machine 10** and **remote machine 30**, *Wookey* provides that the client machine 10 includes an “HTTP client agent,” such as a web browser application, which “can use any type of protocol.” (EX1005 ¶[0159]; *see also id.* ¶[0155].) For example, connection 394 can be made using various protocols, including the Virtual Network Computing (VNC) protocol. (*Id.* ¶[0216]; EX1002 ¶129.) VNC is a graphical desktop-sharing system that allows a computer to remotely access and control another computer. (EX1002 ¶130; EX1022 ¶[0019].) VNC can run over various “industry standard transport protocols, such as TCP/IP, IPX/SPX, NetBEUI.” (EX1005 ¶[0225]; *see also id.* ¶¶[0155], [0215]–[0216]; EX1002 ¶131.)

Wookey further explains that **client machine 10** may use “a different type of protocol than the one used to send the request to the remote machine 30.” (EX1005 ¶[0732]; *see also id.* ¶¶[0738]–[0739], [0772].)

Petition for *Inter Partes* Review
 Patent No. 10,375,215



(*Id.*, FIG. 3D (annotated); *see also id.* ¶[0026]; EX1002 ¶132.)

Wookey also identifies desired characteristics associated at least with the second connection that a POSITA would have considered when choosing a protocol such as negotiating parameters related to the connection, reducing unintentional termination of sessions due to an imperfect connection, detecting and handling disconnections like when a mobile device enters an elevator, and ability to specify inactive time prior to connection termination. (EX1002 ¶¶133–134; EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153]; *see also id.* ¶¶[0738]–[0739]

Petition for *Inter Partes* Review
Patent No. 10,375,215

(disclosing that **remote machine 30'** determines which protocol is being used based on received connection request).)

Wookey further explains that machines **10** and **30'** “exchange a set of messages which negotiate the parameters under which communications will occur.” (EX1005 ¶[0774].) Once negotiations are complete, **client machine 10** and **remote machine 30'**, via the network applications on the machines, “are able to communicate as necessary.” (*Id.*; see also *id.* ¶¶[0744], [0773].) Thus, **client machine 10** and **remote machine 30'** both send and receive packets when establishing a connection. (EX1002 ¶135.) And when using a TCP-based connection for the second connection, something that *Wookey* foresees as noted above, **client machine 10** would necessarily receive a TCP packet from **remote machine 30'** when establishing the second connection by performing a 3-way handshake. (*Id.*)

Moreover, when **client machine 10** uses one of the encoded URLs (e.g., the one pointing to **remote machine 30'**) associated with one of the icons on the HTML page 288, it results in **client machine 10** establishing a connection (e.g., connection 394) with **remote machine 30'** using a transport protocol. (EX1002 ¶136.)

While *Wookey* discloses that various transport protocols (e.g., TCP/IP, IPX/SPX) may be used to establish the second connection and desired characteristics

Petition for *Inter Partes* Review
Patent No. 10,375,215

of such a connection (EX1005 ¶[0225]), *Wookey* does not expressly provide details regarding establishing and maintaining the second connection or the types of negotiable parameters used to establish the second connection. (EX1002 ¶137.) However, it would have been obvious to a POSITA to consider and implement a protocol (which may be “a different type of protocol than the one used to send the request to the remote machine 30” (EX1005 ¶[0732])) that would have met at least some of *Wookey*’s desired characteristics (e.g., such as those identified above) for the second connection between **client machine 10** and **remote machine 30**, and would have resulted in the code, when used by **client machine 10**, resulting in **client machine 10** “receiv[ing] ... a transmission control protocol (TCP)-variant packet” like that recited in limitation 1.d. (EX1002 ¶137.)

Eggert describes such a protocol—a TCP-variant protocol using TCP-variant packets—that allows nodes to negotiate or modify timeout timers on a per-connection basis to provide a reliable and efficient network connection. Accordingly, as discussed below, based on the teachings of *Eggert*, the knowledge of a POSITA, and guidance from *Wookey*, it would have been obvious to configure *Wookey*’s **client machine 10** such that when the “code” (e.g., the encoded URLs in the HTML page 288 or the HTML page 288 including the encoded URLs) is used, **client machine 10** uses the browser application 280 to operate per a protocol such

Petition for *Inter Partes* Review
Patent No. 10,375,215

as the one described by *Eggert* to initiate establishment of the connection between **client machine 10** and **remote machine 30'**, in order to “receive, by the client node, a *transmission control protocol (TCP)-variant* packet.” (EX1002 ¶138.)

(1) *Eggert*

Eggert, like *Wookey*, relates to establishing a reliable connection between two nodes. For instance, where a node may be mobile, the node may experience “changes [in] network attachment points based on [its] current location”, and thus *Eggert* discloses networking features in the same technical field as *Wookey*. (EX1006, 1–3; EX1002 ¶¶139–141; *infra* Section IX.A.1.d.2 (discussing similarities between *Wookey* and *Eggert*).)

Eggert describes a modification to the TCP protocol to “allow established TCP connections to survive periods of disconnection” (EX1006, Abstract). In particular, *Eggert* introduces a “TCP Abort Timeout Option” (ATO) that “allows conforming TCP implementations to negotiate individual, per-connection abort timeouts.” (*Id.*) Thus, as explained below, *Eggert* discloses a TCP-variant protocol that allows established TCP connections to survive periods of disconnection by negotiating abort timeouts. (*Id.*; EX1002 ¶142; *see also id.* ¶¶82–91 (*Eggert* overview).)

Petition for *Inter Partes* Review
Patent No. 10,375,215

Eggert explains that its TCP-variant protocol solves the problem “where hosts are only intermittently connected to the Internet.” (EX1006, 1–3.) For example, a mobile host may “experience disconnected periods during which no network service is available” when “mobile hosts ... change network attachment points.” (*Id.* 3.) These disconnected periods may lead to the “established TCP connections” being “abort[ed] during periods of disconnection.” (*Id.*) *Eggert* overcomes this problem in the art by allowing hosts to negotiate per-connection abort timeouts. (*Id.*) *Eggert*’s TCP-variant protocol “allow[s] mobile hosts to maintain TCP connections across disconnected periods that are longer than their system’s default abort timeout.” (*Id.*; EX1002 ¶¶143–144.)

Eggert further explains that if “[a] TCP implementation” on a device “does not support the TCP [ATO],” then the device “SHOULD silently ignore it.” (EX1006, 7.) *Eggert*, thereby, provides a new TCP option to the traditional TCP implementation where devices configured according to *Eggert* vary from the “traditional TCP implementation” in terms of the ATO negotiation. (EX1002 ¶¶145–146.) And as discussed below, just like the ’215 patent, *Eggert*’s ATO is also incorporated in the TCP header and arranged in a similar format. (*Compare* EX1001, 15:52–60, FIG. 8 *with* EX1006, 3–5, FIG. 1; EX1002 ¶147.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

Thus, a connection using *Eggert*'s ATO would be a TCP-variant connection and at least the segments used for establishing such a TCP-variant connection (e.g., the segments using the TCP ATO) would be TCP-variant packets. (*See also supra* Section VIII; EX1002 ¶148.)

Eggert's TCP-variant protocol uses the three-way handshake to establish a connection between two nodes on a network. (EX1006, 5; EX1002 ¶149.) A three-way handshake involves the exchange of synchronize (SYN) and acknowledge (ACK) messages between the nodes to create a connection—i.e., SYN, SYN-ACK, and ACK message in a standard TCP implementation. (EX1002 ¶149; EX1008, 31–32, 34–37, *see also* EX1001, 14:57–15:3 ('215 patent disclosing the standard TCP three-way handshake).)

As discussed below and exemplified in Figure 2 below, *Eggert* discloses the features of this limitation in two different ways such that that a first node receives a TCP-variant packet during the exchange of three messages (3-way handshake) in *Eggert*'s TCP-variant protocol for negotiating a per-connection abort timeout.¹⁴

¹⁴ In the combined *Wookey-Eggert* process/system discussed below (e.g., *infra* Section IX.A.1.d.2), the combined process/system discloses the claim limitations

Petition for *Inter Partes* Review
 Patent No. 10,375,215

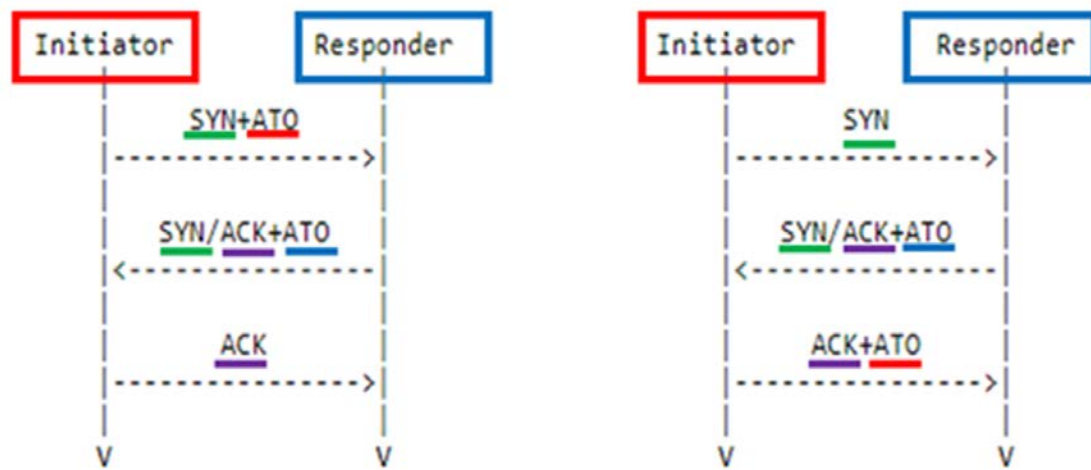


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

(EX1006, FIG. 2 (annotated); EX1002 ¶150.)

First, as shown on the left-hand side of Figure 2 (when the **initiator** initiates an abort timeout negotiation), *Eggert* teaches that the **initiator** (e.g., client node) sends the **responder** (e.g., server node) a “**SYN+ATO**” segment.¹⁵ (EX1006, FIG.

(e.g., limitations 1.d–g) under both embodiments shown in *Eggert* (i.e., left and right side of Figure 2).

¹⁵ The '215 patent describes a “packet” in a manner consistent with what the TCP standard (RFC 793) describes as “segment” at that time. (Compare EX1001, 15:31–41, FIG. 8 with EX1008 (RFC 793), 15, FIG. 3 (showing the same TCP header in a

Petition for *Inter Partes* Review
Patent No. 10,375,215

2; *see also id.*, 5–7.) The ATO in the segment indicates that this segment contains an abort timeout. (*Id.*, 3.) *Eggert* explains that when the responder (e.g., server node) accepts the initiator’s offered abort timeout, it must echo the offered timeout value in the ATO it sends. (EX1006, 5–7.) That is, the initiator receives an “SYN/ACK+ATO” segment (“TCP-variant packet”) from the responder. (*Id.*; EX1002 ¶¶151–152.)

Second, as shown on the right-hand side of Figure 2 (when the responder initiates an abort timeout negotiation), *Eggert* teaches that the initiator (client node) receives an “SYN/ACK+ATO” segment (“TCP-variant packet”) from a responder (server node). (EX1006, 5–7; EX1002 ¶153.)

The received (SYN/ACK+ATO) segment in both scenarios (left and right side of Figure 2) is a TCP-variant packet because it contains an abort timeout option (ATO), which is an option not standard in the TCP protocol as defined by RFC 793. (EX1002 ¶154.) *See also* Google -845 IPR, Paper 16 at 20–21 (Board agreeing that *Eggert* discloses a TCP-variant packet for similar reasons).

TCP “packet” (EX1001) and TCP “segment” (EX1008)).) Thus, *Eggert*’s TCP “segment” is the same as the TCP “packet” as described and claimed in the ’215 patent. (EX1002 ¶151.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

Indeed, *Eggert*'s TCP ATO shown in Figure 1 uses the same TCP structure as the idle time period (ITP) field illustrated in Figure 8 of the '215 patent. For example, like the '215 patent, *Eggert*'s ATO is also incorporated in the TCP header and arranged in a similar format. (*Compare* EX1001, 15:52–60, FIG. 8 with EX1006, 3–5, FIG. 1.) As shown below, both formats use a KIND sub-field to indicate the new option, a TCP option length sub-field, and a data portion (YELLOW) of the TCP option containing the timeout value. (*Compare* EX1001, FIG. 8 with EX1006, FIG. 1.)

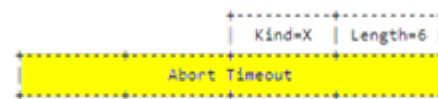
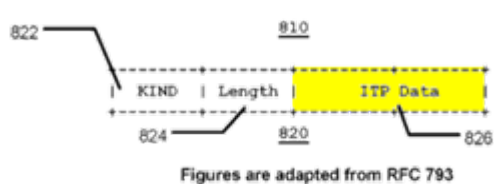


Figure 1: TCP Abort Timeout Option

(EX1001, FIG. 8 (annotated) (left); EX1006, FIG. 1 (annotated) (right); EX1002 ¶¶147–148.) Thus, *Eggert* uses the same format for the ATO as the embodiment disclosed in the '215 patent for a TCP-variant packet. (EX1002 ¶155.)

The understanding that *Eggert*'s “SYN/ACK+ATO” segment is a TCP-variant packet in the context of the '215 patent is also consistent with PO's proposed construction of “variant” in a related matter. (*See, e.g.*, EX1018, 60 (Exhibit C) (PO defining a “TCP-variant as one that “manifest[s] variety, deviation, or disagreement,

Petition for *Inter Partes* Review
Patent No. 10,375,215

[or] var[ies] slightly from the [TCP] standard.”).) *Eggert* describes a TCP-variant packet under PO’s interpretation because the packets exchanged during the three-way handshake in *Eggert* vary, as described above, from those exchanged in the TCP standard.

Eggert thus discloses receiving, by a client node, a transmission control protocol (TCP)-variant packet, like that claimed. (EX1002 ¶156.)

(2) Reasons to Combine

Based on *Wookey*’s and *Eggert*’s disclosures, a POSITA’s knowledge, and the discussions above, it would have been obvious to configure and implement *Wookey*’s **client machine 10** to use a TCP-variant protocol like the one disclosed by *Eggert* when establishing (and using) the second connection between **client machine 10** and **remote machine 30’** concerning claim limitations 1.d–1.g. Such an implementation would have had many advantages and been a predictable and straightforward result achieved by combining well-known technologies using known methods. (EX1002 ¶157.) *See KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 416–18 (2007).

Eggert and *Wookey* disclose features in a similar technological field. (EX1002 ¶158.) For example, both *Wookey* and *Eggert* are concerned about maintaining connections in situations where at least one node (e.g., a mobile node)

Petition for *Inter Partes* Review
Patent No. 10,375,215

in the connection may lose connectivity due to its movement. (*Id.*) *Wookey* discloses that “client machine 10 may traverse network segments or network access points that cause changes in the network address ... or causes the client machine 10 to disconnect.” (EX1005 ¶[0581]; *see also id.*, ¶¶[1135]–[1136] (discussing the need for connections to manage unintentional disconnection between the client and remote machines), [1153] (discussing an unintentional disconnection when **client machine 10** enters an elevator).) Similarly, *Eggert* discloses that “[l]engthening abort timeouts allows established TCP connections to survive periods of disconnection” as “[s]ome hosts are only intermittently connected to the Internet.” (EX1006, 1–3.) Like *Wookey*, *Eggert* discloses that “[o]ne example [of such unintentional disconnections] is mobile hosts that change network attachment points based on current location” and “[i]n between connected periods, mobile hosts may experience disconnected periods during which no network service is available.” (*Id.*, 2–3.) Thus, both *Wookey* and *Eggert* are directed to establishing reliable connections between two nodes, where the nodes may be mobile nodes that experience an unintentional disconnection, so a POSITA would have had reason to consider *Eggert* when contemplating and implementing *Wookey*’s teachings. (EX1002 ¶¶159–161.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

Moreover, as explained above, *Wookey* describes providing access to code that, when used by **client machine 10**, causes **client machine 10** to establish a connection between **client machine 10** and **remote machine 30'** using a protocol that can be different from the one used to connect **client machine 10** and **remote machine 30**. *Wookey* further discloses the types of characteristics a POSITA would have considered concerning the protocol that can be used for the second connection, such as negotiating parameters related to the connection, reducing unintentional termination of sessions due to an imperfect connection, detecting and handling disconnections like when a mobile device enters an elevator, and ability to specify a permissible inactive time before connection termination. (EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153]; *see supra* Section IX.A.1.d.) A POSITA would have been thus motivated by *Wookey*'s disclosures and suggestions to consider protocols that would provide these desired features for the connection with **remote machine 30'**, including the features associated with the TCP-variant connection described by *Eggert*. Such a POSITA would have been motivated to incorporate a protocol like *Eggert*'s TCP-variant protocol in the use of the “code” by **client machine 10** for establishing a connection with **remote machine 30'**. (EX1002 ¶¶162–163.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

Using the TCP variant protocol with an ATO option as described in *Eggert* would have provided **client machine 10** with benefits including an improved, reliable connection to **remote machine 30'** with the versatility of negotiated timeout parameters consistent with *Wookey's* desired characteristics for the second connection. *See KSR*, 550 U.S. at 416. Indeed, by using a TCP variant protocol with an ATO option (or similar feature as described by *Eggert*), *Wookey's* method could “negotiate individual, per-connection abort timeouts” (EX1006, 1) “to survive periods of disconnection” (*id.*) over a reliable connection. (EX1002 ¶164.) Thus, *Wookey's* method for “providing a client with a reliable connection to a host service” (EX1005 ¶[0077]) would have been improved by allowing “[l]ong abort timeout values” so that the “hosts ... tolerate extended periods of [temporary] disconnection,” as provided by *Eggert* (EX1006, 5). (EX1002 ¶164.) Moreover, configuring *Wookey* to utilize a TCP-variant protocol based on *Eggert's* teachings would have been both a predictable and straightforward implementation. (EX1002 ¶165; EX1005 ¶[0216].); *see KSR*, 550 U.S. at 416.

Additionally, *Eggert's* protocol aims to avoid or mitigate interruptions caused by intermittent disconnections between the nodes, a problem recognized by *Wookey*. (EX1002 ¶167; EX1006, 1–3; EX1005 ¶[0903].) Thus, a POSITA would have been motivated to consider *Eggert's* features when considering how to implement

Petition for *Inter Partes* Review
Patent No. 10,375,215

Wookey's process/system and modify *Wookey* as discussed. Indeed, *Eggert*'s TCP variant protocol, which allows the nodes to negotiate an abort timeout, would have improved *Wookey*'s process just as described in *Eggert*. (EX1002 ¶167.) Accordingly, such a combination would have yielded the predictable result of providing a network communication exchange that minimizes disruptions caused by timeout issues. (*Id.*); see *KSR*, 550 U.S. at 417.

A POSITA would have further recognized that the *Wookey-Eggert* combination would have involved the use of known technologies (e.g., aspects of similar protocols) and design concepts and processes to obtain the foreseeable result of a reliable connection between **client machine 10** and **remote machine 30**'. (EX1002 ¶168). The above-modification would have involved substituting features from one reliable protocol from a finite number of available alternative reliable communication protocols, such as that described by *Eggert*. (*Id.*) Thus, a POSITA would have had a reasonable expectation of success in the above-modification. (*Id.*) See *Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1364 (Fed. Cir. 2007) ("only a

Petition for *Inter Partes* Review
Patent No. 10,375,215

reasonable expectation of success, not a guarantee, is needed” in an obviousness analysis).¹⁶

A POSITA would thus have found configuring *Wookey*’s system/process such that the “code” (e.g., encoded URLs in HTML page 288 or page 288 including the encoded URLs), when used by **client machine 10**, would cause **client machine 10** to utilize a TCP-variant protocol like *Eggert*’s TCP-variant protocol a foreseeable and straightforward implementation. (EX1002 ¶166); see *KSR*, 550 U.S. at 416. For example, as explained, *Wookey* discloses that “*the URL includes a file, or a reference to a file, that contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (EX1005 ¶[0216].) In light of the reasons discussed above, a POSITA would have been motivated to configure *Wookey*’s encoded URL such that it includes information causing **client machine 10** to establish a connection with **remote machine 30’** using the TCP-variant protocol (like that described by *Eggert*). (EX1002 ¶166.) This

¹⁶ There is no evidence of nonobviousness at this time that would overcome the overwhelming evidence of obviousness as presented herein. Petitioner reserves the right to address any such evidence/argument should PO later present any in this proceeding.

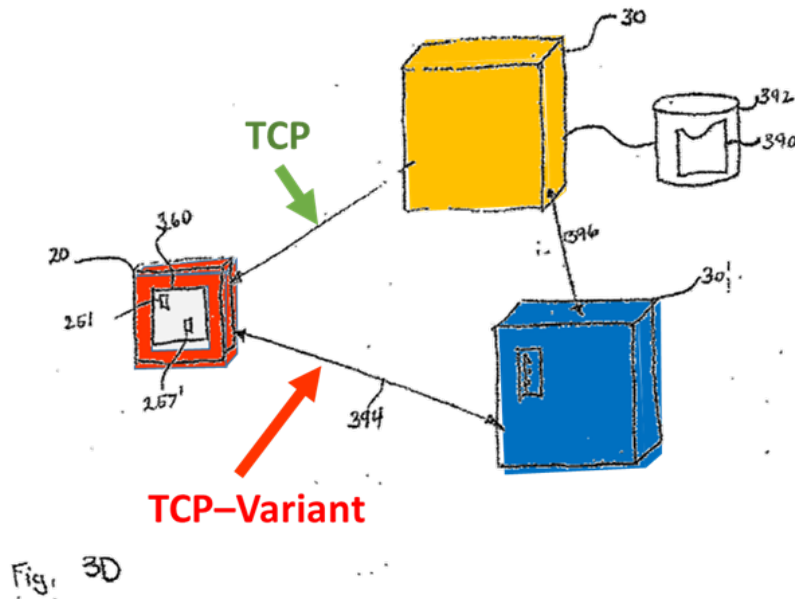
Petition for *Inter Partes* Review
Patent No. 10,375,215

implementation could have been achieved by combining well-known technologies using known methods, such as known network design concepts and technologies described by *Wookey* and *Eggert* and known in the art at the time. (*Id.*)

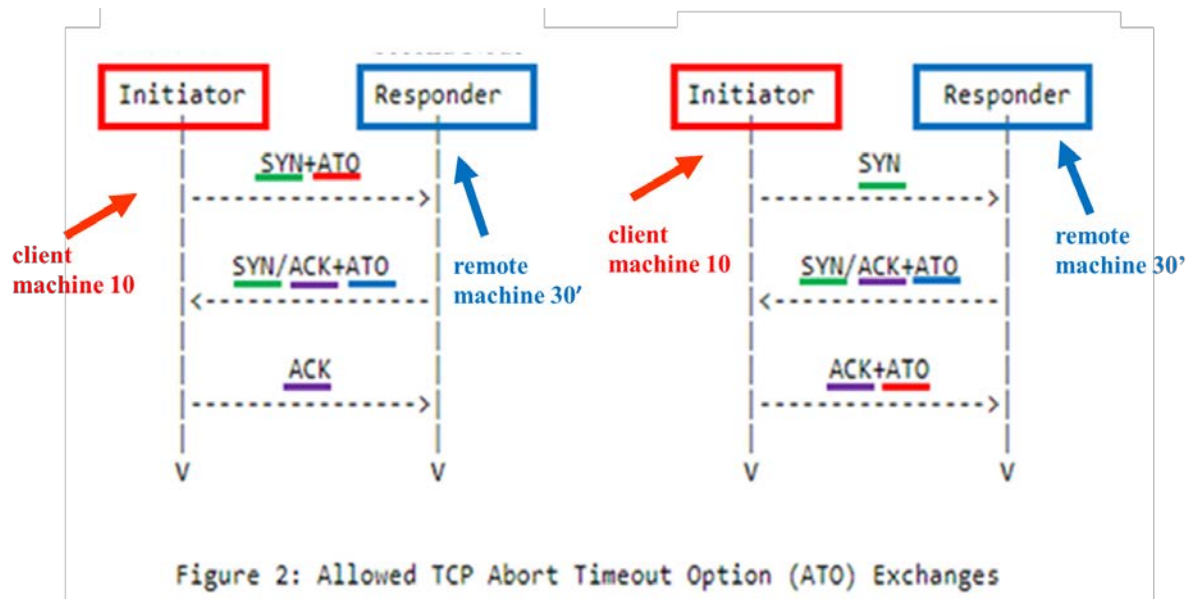
Indeed, a POSITA would have been skilled and would have had the knowledge to configure *Wookey*'s system and method to implement a TCP-variant packet and protocol in various ways, while taking into account any known programming, design, and other related concepts, limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. For example, a POSITA would have been motivated based on such disclosures to configure *Wookey*'s system and process consistent with the exemplary annotated figures from *Wookey* and *Eggert* below.¹⁷

¹⁷ The configurations discussed below are exemplary and not limiting. (EX1002 ¶170.)

Petition for *Inter Partes* Review
 Patent No. 10,375,215



(EX1005, FIG. 3D (annotated); EX1002 ¶¶168–169.)



(EX1006, FIG. 2 (annotated); EX1002 ¶¶168–169.)

Petition for *Inter Partes* Review
 Patent No. 10,375,215

Accordingly, the combination of *Wookey* and *Eggert* discloses and/or suggests limitation 1.d. (EX1002 ¶171.)

e) **“detect an idle time period parameter field in the TCP-variant packet,”**

The *Wookey-Eggert* combination discloses and/or suggests this limitation. As discussed above for claim limitation 1.d, the combined *Wookey-Eggert* system/process would have involved negotiating the abort timeout option included in the received “SYN/ACK+ATO” segment (“TCP-variant packet”) at **client machine 10** for establishing a TCP-variant connection with **remote machine 30’**. (*Supra* Section IX.A.1.d; EX1002 ¶¶172–181.)

Eggert explains that this negotiation includes the **initiator** detecting the ATO field (“idle time period parameter field”) in the “SYN/ACK+ATO” segment (“TCP-variant packet”).

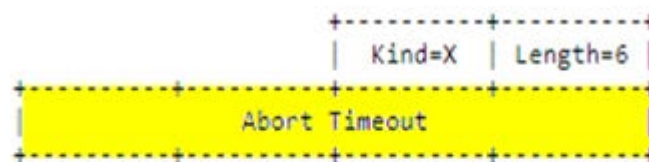


Figure 1: TCP Abort Timeout Option

(EX1006, FIG. 1 (annotated); EX1002 ¶174.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

First, regarding the left-hand side of Figure 2, *Eggert* provides that the **initiator** must accept the **responder's** returned abort timeout value because the **initiator** proposed the abort time in the initial “**SYN+ATO**” segment it sent to the **responder**. (EX1006, 5–7 (explaining that “[t]he host that initially proposed the Abort Timeout Option analyzes the next segment it receives from its peer” and if the next segment contains the ATO, “the connection MUST use the abort timeout contained inside the Abort Timeout Option.”); EX1002 ¶175.)

Second, regarding the right-hand side of Figure 2, the **initiator** may either accept or shorten the offered abort timeout from the **responder**. (EX1002 ¶176.)

Thus, for either side of Figure 2, the **initiator** (e.g., client node) must first detect the ATO parameter field containing the ATO value before it can accept or shorten the value. (EX1006, 5.) In this way, *Eggert* discloses detecting an ATO field in the TCP-variant packet. (EX1002 ¶176.)

The ATO field represents an idle time period parameter field used to convey an abort timeout value between two hosts. (EX1006, 1–7, FIG. 1.) The ATO field contains the abort timeout value, which defines an “idle time period,” as that claimed. (EX1006, 2–5, FIG. 1.) For example, the abort timeout value in the ATO field “is the desired abort timeout of the connection, specified in seconds.” (*Id.*, 3, FIG. 1; *see also id.*, Abstract.) *Eggert* further explains that by “[l]engthening [the]

Petition for *Inter Partes* Review
Patent No. 10,375,215

abort timeout” value that the “established TCP connections” can “survive periods of disconnection.” (*Id.*, Abstract.) The abort timeout value defines an idle time period because it is associated with a period where no packet is communicated in the connection to keep the connection active. (EX1002 ¶¶177–179; *see also* EX1006, Abstract; *see infra* Section IX.A.1.f.)

Accordingly, for reasons similar to those explained above, the *Wookey-Eggert* combined process would have been further configured to include the process of detecting, by **client machine 10**, an ATO field (“idle time period parameter field”) in the “SYN/ACK+ATO” segment (“TCP-variant packet”) received from **remote machine 30**. (See *supra* Section IX.A.1.d (discussing the *Wookey-Eggert* combination).) Accordingly, the *Wookey-Eggert* combination discloses and/or suggests limitation 1.e. (EX1002 ¶¶180–181.)

- f) **“identify metadata in the idle time period parameter field for an idle time period, during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active, and”**

The *Wookey-Eggert* combination discloses and/or suggests this limitation. (EX1002 ¶¶182–195.) As discussed below, *Eggert* discloses identifying an abort timeout value (“metadata”) in the abort timeout field (“idle time period parameter field”) for an abort timeout (“idle time period”), during which no packet is

Petition for *Inter Partes* Review
Patent No. 10,375,215

communicated in a TCP-variant connection to keep the TCP-variant connection active. (*Id.*; *see also supra* Section IX.A.1.e.)

Eggert discloses that the **initiator** (client node) “identif[ies] metadata in the idle time period parameter field for an idle time period,” as claimed. For example, as discussed above for claim limitation 1.e, *Eggert* explains that the **initiator** accepts or shortens the offered abort timeout value (“metadata”) specifying an abort timeout (“idle time period”) contained within the “SYN/ACK+ATO” segment (“TCP-variant packet”). (*Supra* Section IX.A.1.e; EX1006, 5–7.) Thus, the **initiator** must first identify the abort timeout value before it can either accept or shorten the offered abort timeout. (EX1006, 5–7; EX1002 ¶184.) Consequently, *Eggert* discloses that the **initiator** processes the received “SYN/ACK+ATO” segment from the **responder** according to an established format (EX1006, FIG. 1) to identify which bits of the option represents the proposed abort timeout duration value. (*Id.*, 3–5; *see also* EX1002 ¶184.)

Eggert explains that the “Abort Timeout” is “specified in seconds” (EX1006, 3) and is represented by a “32-bit value” (*id.*, 9), and thus the bits that identify the ATO field as associated with an abort timeout (KIND, LENGTH) and the bits representing the abort timeout value each constitute “metadata” for the proposed or responsive abort timeout (“idle time period”). (EX1002 ¶185.) Further, as discussed

Petition for *Inter Partes* Review
Patent No. 10,375,215

above for claim limitation 1.e, the format for representing the TCP abort timeout in Figure 1 of *Eggert* is similar to the ITP data field 826 in Figure 8 of the '215 patent. (*Id.*; *supra* Section IX.A.1.e.)

The manner as to how *Eggert* discloses identifying such values is consistent with the '215 patent's discussions. For example, the '215 patent explains that the metadata can be a "duration of time" (EX1001, 16:1–5, 22:2–5), which is what the ATO value in *Eggert* contains (EX1006, 3–5, FIG. 1; EX1002 ¶186.) Moreover, the understanding that *Eggert*'s descriptions regarding identifying values, such as the abort timeout value, discloses identifying metadata like that claimed is consistent with PO's view of the same claimed features. (EX1017, 7 (PO alleging "identify[ing] metadata (e.g., a value, etc.) in the idle time period parameter field (e.g., idle timeout parameter field, etc.) for an idle time period"), ("the metadata includes a value in seconds").)

Eggert further discloses that the timeout value in the ATO field corresponds to an idle time period, "during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active," as claimed. (EX1002 ¶187.) For example, *Eggert* explains that the abort timeout duration specified in the ATO field is used to track periods during which there is no communication to keep the connection active. (EX1006, 1–7; *see also id.*, 9 ("Long abort timeout values allow

Petition for *Inter Partes* Review
Patent No. 10,375,215

hosts to tolerate extended periods of disconnection”).) A POSITA would have recognized that the packet that must be “communicated ... to keep the ... connection active” in *Eggert* is an ACK that a node receives in response to a previous sent segment. (EX1002 ¶188.) Indeed, *Eggert*’s ATO relies upon “[t]he TCP specification [1] [which] includes a ‘user timeout’ that defines the maximum amount of time that segments may remain *unacknowledged* before TCP will abort the connection. If a disconnection lasts longer than the user timeout, the TCP connection will abort.” (EX1006, 3 (emphasis added).) Like *Eggert*, the ’215 patent also describes modifying a TCP user timeout. (*Compare* EX1001, 13:54–57, 22:23–30 *with* EX1006, 3.) Accordingly, *Eggert*’s description of the abort timeout as the duration of time during which a node can wait to receive an ACK is a period “during which no packet is communicated in the TCP-variant connection to keep the TCP-variant connection.” (EX1002 ¶¶188–190.)

Disconnection

The combined *Wookey-Eggert* process/system discloses or suggests that the timeout value in the ATO field corresponds to an idle time period, “during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active,” as claimed, in an **additional** way where a disconnection occurs

Petition for *Inter Partes* Review
Patent No. 10,375,215

between **client machine 10** and **remote machine 30**'.¹⁸ (EX1005 ¶¶[0581], [1135]-[1136], [1153]; EX1006, 1–3; *see also supra* Section IX.A.1.d.2 (noting that both *Wookey* and *Eggert* aim to maintain connection between nodes through unintentional disconnection periods); EX1002 ¶191.)

For example, when an interruption in communication occurs such as a disconnection in the network service (as contemplated by *Wookey* and *Eggert*) or in a physical network medium (as described in the '215 patent) (EX1001, 2:11–14, 2:28–31, 12:67–13:6), packets from one node do not reach the other node in either direction. (EX1002 ¶192.) In this situation contemplated by the '215 patent, no packets are communicated between the nodes to keep the connection active because they cannot get through the interruption or the physical medium's disconnection. (EX1002 ¶192; EX1001, 12:64–13:6, 2:28–31, 22:22–30, FIG. 7 (annotated below in yellow showing no communication between the nodes during a disconnection).)¹⁹

¹⁸ This is consistent with how the '215 patent describes such features. (EX1001, 12:64–13:6, 2:28–31, 22:22–30 (describing that the idle time period may be linked to the physical disconnection of a network medium or simply a dead connection).)

¹⁹ The '215 patent repeatedly confirms that in order for a packet to be “communicated ... to keep the ... connection active,” a packet sent by one node must

Petition for *Inter Partes* Review
Patent No. 10,375,215

Indeed, neither node would have received any packets from the other node during such a disconnection period. Therefore, neither node would have attempted to send a responsive acknowledgment packet during the period. (EX1002 ¶192.) This is the type of situation where, as discussed above, *Eggert* aims to maintain the connection, e.g., maintain the connection even when no packet is communicated during the negotiated abort timeout period, as there would be no received packets to acknowledge. *Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016) (“[C]ombinations of prior art that sometimes meet the claim elements are

be received by the other node. (EX1002 ¶193; EX1001, 11:41–46 (“With reference to the method illustrated in FIG. 2, block 202 illustrates the method includes receiving, by a first node, first idle information for detecting a **first idle time period during which no TCP packet** including data in a first data stream **sent** in the TCP connection **by a second node** is *received* by the first node.”) (emphasis added), 11:46–51 (disclosing that during the idle time period, no TCP packet “**sent** in the TCP connection **by a second node** is *received* by the first node”), 11:51–57 (same), 21:18–37 (same), 19:64–20:6 (disclosing that during the idle time period, “**no TCP packet in the TCP connection is *received***, by the first node 602 ... from second node 604”), FIG. 2 (block 202), FIG. 3 (block 304), FIGS. 6–7.)

Petition for *Inter Partes* Review
 Patent No. 10,375,215

sufficient to show obviousness.”); *Power Integrations, Inc. v. Fairchild Semiconductor Int'l, Inc.*, 843 F.3d 1315, 1336-37 (Fed. Cir. 2016) (same).

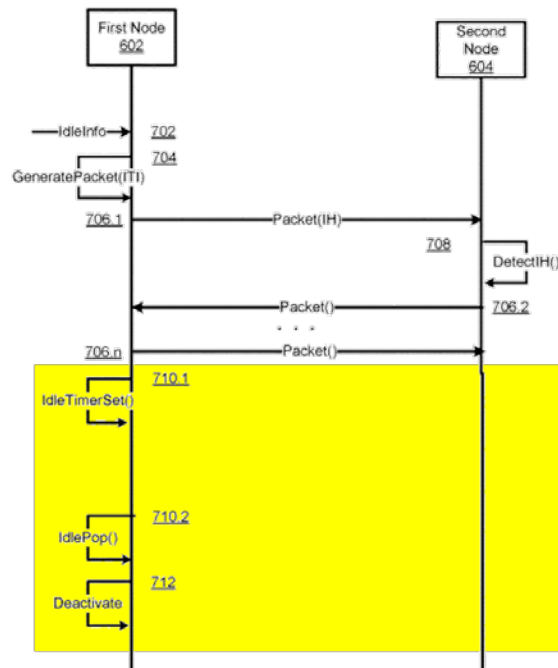


Fig. 7

(EX1001, FIG. 7 (annotated); EX1002 ¶192.)

* * *

Accordingly, for reasons similar to those explained above, the *Wookey-Eggert* combined process would have been further configured to include the process of **client machine 10** identifying a value (“metadata”) in the abort timeout field (“idle time period parameter field”) for an idle time period, during which no packet is communicated in a TCP-variant connection to keep the TCP-variant connection

Petition for *Inter Partes* Review
Patent No. 10,375,215

between **client machine 10** and **remote machine 30'** active. (*See supra* Sections IX.A.1.d–1.e.) Thus, the *Wookey-Eggert* combination discloses and/or suggests limitation 1.f. (EX1002 ¶¶194–195.)

- g) **“determine, based on the metadata, a timeout attribute associated with the TCP-variant connection.”**

The *Wookey-Eggert* combination discloses and/or suggests this limitation. (EX1002 ¶¶196–204.) As discussed below, the *Wookey-Eggert* combination discloses and/or suggests determining, based on the abort timeout value (“metadata”), a timeout attribute associated with the TCP-variant connection in at least two ways. (*Id.*; *see also supra* Sections *Supra* Section IX.A.1.d–f.)

For instance, *Eggert* discloses that while “[m]any TCP implementations default to user timeout values of a few minutes” (EX1006, 3), after successful negotiation, hosts use the abort timeout for the connection. (*Id.*, 7 (“If the next reply segment does not contain an Abort Timeout Option, the connection MUST use the default abort timeout. If it does, the connection MUST use the abort timeout contained inside the Abort Timeout Option.”).) Figure 2 discloses the two scenarios for ATO negotiation, both of which disclose claim limitation 1.g.

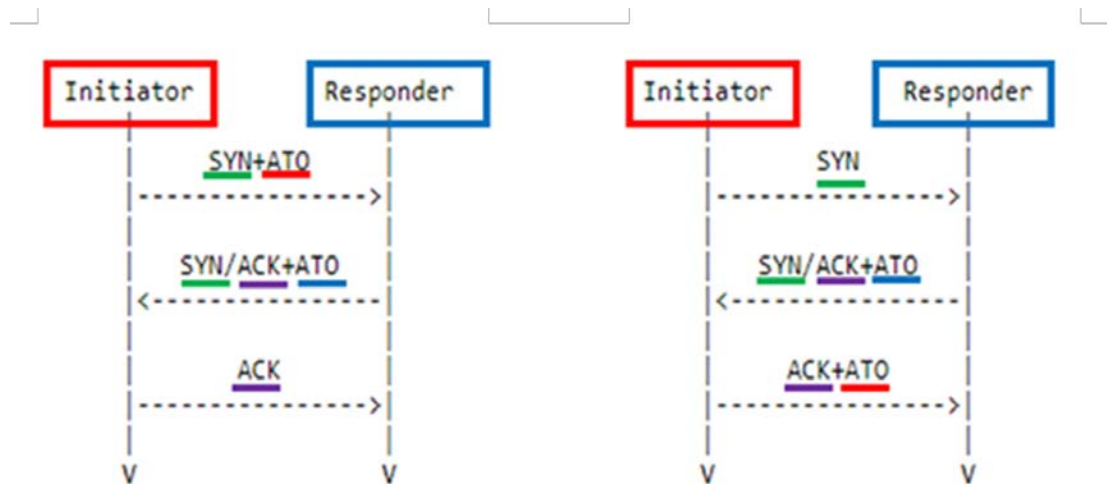


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

(*Id.*, FIG. 2 (annotated); EX1002 ¶198.)

First, *Eggert* explains that the **initiator** in the left-hand side scenario of Figure 2 has only a single option because it proposed the abort timeout value. (EX1006, 5–7.) Namely, the **initiator** “MUST be prepared to accept a shorter timeout value [from the **responder**] than proposed after the negotiation.” (*Id.*, 5.) *Eggert*’s disclosure of the **initiator**’s acceptance of the timeout value from the **responder** for the TCP-variant connection describes a “determine” process like that recited in limitation 1.g. (EX1002 ¶199.)

Second, as shown on the right-hand side of Figure 2, the **initiator**, because the **responder** specified the initial ATO in the “**SYN/ACK+ATO**” segment, may decide “whether to accept, shorten, or reject its peer’s proposed abort timeout.”

Petition for *Inter Partes* Review
Patent No. 10,375,215

(EX1006, 5–7; *see also id.*, 9.) Thus, the **initiator** determines a timeout attribute, that is, the abort timeout for the connection, based on accepting, shorting, or rejecting the value (“metadata”) in the ATO field in the “SYN/ACK+ATO” segment. (EX1002 ¶200.)

The ’215 patent describes timeout attributes broadly and is consistent with *Eggert*’s abort timeout value. (EX1001, 22:23–28 (“ITP option handler component 562 may [*sic*] one or more attribute option handler components 564 to modify one or more corresponding attributes of a keep-alive option, a TCP user timeout, a retransmission timeout, an acknowledgment timeout, and another timeout associated with the TCP connection, in response to identifying the ITP header”); EX1002 ¶201.)

Accordingly, for reasons similar to those explained above, the *Wookey-Eggert* combined process would have been further configured to include the process of **client machine 10** determining, based on the abort timeout value (“metadata”), a timeout attribute associated with the TCP-variant connection. (*See supra* Sections IX.A.1.d–1.f.) Thus, the *Wookey-Eggert* combination discloses and/or suggests limitation 1.g. (EX1002 ¶¶202–204.)

B. Ground 2: The Combination of *Wookey*, *Eggert*, and *Abdolbaghian* Render Obvious Claims 4, 8, and 9

1. Claim 4

Wookey in combination with *Eggert* and *Abdolbaghian* discloses and/or suggests the limitations of claim 4. (EX1002 ¶¶205–234.)

- a) **“The method of claim 1 wherein the code, to which the access is provided, is configured such that, in response to being used by the client node, causes the client node to operate such that one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.”²⁰**

As discussed above for claim 1, in the combined *Wookey-Eggert* process/system, the code (e.g., the encoded URLs in the HTML page 288 or the HTML page 288 including the encoded URLs) is configured such that, in response to being used by **client machine 10**, causes **client machine 10** to initiate the process for establishing a TCP-variant connection with **remote machine 30’**. (*Supra* Sections IX.A.1.c–g; EX1002 ¶¶206–207.)

First, as discussed below, the combined *Wookey-Eggert* process/system discloses and/or suggests that the code would have caused **client machine 10** “to operate such that one or more keep-alive packets are communicated, based on a

²⁰ See *supra* n.11.

Petition for *Inter Partes* Review
Patent No. 10,375,215

keep-alive period,” as claimed. Second, as discussed below, while the combined *Wookey-Eggert* process/system does not explicitly disclose that the keep alive period is “based on the idle time period,” as claimed, it would have been obvious to set the keep alive period in the *Wookey-Eggert* process/system based on the idle time period in light of a knowledge of a POSITA and the teachings of *Abdolbaghian*. (EX1002 ¶208.)

Regarding the first point, *Wookey* discloses that **client machine 10** periodically communicates one or more keep-alive packets to maintain the connection. (EX1002 ¶209.) For example, *Wookey* explains that “client machine 10 periodically transmits a signal to the remote machine 30 to confirm that a connection is still intact.” (EX1005 ¶[1136].)²¹ “[I]f the server process 7922 detects

²¹ While *Wookey*’s discussion regarding periodic transmission of packets to confirm that the connection is intact refer to a connection between **client machine 10** and **remote machine 30** (EX1005 ¶¶[1134]–[1136]), **client machine 10** would have operated in a similar fashion when connected to **remote machine 30**’ in the *Wookey-Eggert* combination given that *Eggert* simply provides the added functionality of negotiating abort timeout values on a per-connection basis when establishing the second connection in *Wookey*’s disclosed processes. (EX1002 ¶209 n.11.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

that a predetermined number of expected confirmation signals from a client machine 10 have not arrived, the server process 7922 determines that the client machine 10 has disconnected.” (*Id.*) Upon detecting this disconnection, “the server continues execution of an application 7916 for a fixed time period, and if a user fails to connect within that time period, the server process 7922 stalls the application 7916” (*id.* ¶[1135]) or the server may “disconnect an application session 7918 from the client machine 10 that the user is communicating from” (*id.* ¶[1136].)

Wookey’s disclosure of periodically transmitting signals to confirm that the connection is still intact is a “keep-alive” mechanism for communicating keep-alive packets based on a keep-alive period as was known in the art at the time. (EX1002 ¶210; EX1009, 101–102; EX1011, 54.) In order to communicate the signals (“keep-alive packets”) “periodically” as described by *Wookey*, **client machine 10** would have maintained a keep-alive period (e.g., a timer) based on which the signals were periodically communicated. (EX1005 ¶[1136]; EX1002 ¶211; EX1011, 54; EX1009, 101.)

Eggert similarly discloses that the ATO allows the nodes to negotiate how long each node should keep the connection alive during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active. For example, *Eggert* explains that the ATO “allows mobile hosts to maintain

Petition for *Inter Partes* Review
Patent No. 10,375,215

TCP connections across disconnected periods.” (EX1006, 2.) That is, *Eggert* teaches “[l]engthening abort timeouts allows established TCP connections to survive periods of disconnection.” (EX1006, Abstract.) The ’215 patent itself acknowledges that the “TCP keep-alive option is supported by a number of implementations of the TCP” (EX1001, 1:54–56) and advantages of the keep-alive option include “detecting a dead peer/partner endpoint sooner,” “indirectly detecting when a network is so congested that two nodes with endpoints in a TCP connection are effectively disconnected,” and “keep[ing] an inactive TCP connection open” (*id.*, 2:10–16). (EX1002 ¶212.)

In light of such disclosures and knowledge of a POSITA, and in context of the above-discussed *Wookey-Eggert* combination (*supra* Section IX.A.1), the code in the combined process/system would have been configured to cause **client machine 10** to keep the TCP-variant connection alive for periods of inactivity based on the timeout value negotiated via the ATO, along with communication of keep-alive signals based on a keep-alive period to maintain the connection between **client machine 10** and **remote machine 30**. (EX1002 ¶213; *see also* Sections IX.A.1.d.2 (regarding reasons for combining *Wookey* and *Eggert* applicable here).) Thus, for reasons similar to those discussed above, the *Wookey-Eggert* combination discloses and/or suggests that the code would have caused **client machine 10** “to operate such

Petition for *Inter Partes* Review
Patent No. 10,375,215

that “one or more keep-alive packets are communicated, based on a keep-alive period.” (EX1002 ¶214.)

With respect to the second point, while the combined *Wookey-Eggert* system/process does not expressly disclose that the “keep-alive period” is “based on the idle time period,” as claimed, it would have been obvious to set the keep alive period in the *Wookey-Eggert* process/system based on the idle time period in light of a knowledge of a POSITA and the teachings of *Abdolbaghian*, as discussed below. (EX1002 ¶215.)

(1) *Abdolbaghian*

Abdolbaghian, like *Wookey* and *Eggert*, relates to managing client/server connections in a networked environment. (EX1002 ¶216; *see also* EX1007, 1:14–28, 1:49–50, 3:1–6, FIG. 1, cl. 1.) For example, *Abdolbaghian* relates to a communication network 4 comprised of connected computers 3a–c and the user terminals 1a–c. (EX1007, 2:8–15, FIG. 1; EX1002 ¶216; *see also id.* ¶¶92–99 (*Abdolbaghian* overview).)

Petition for *Inter Partes* Review
 Patent No. 10,375,215

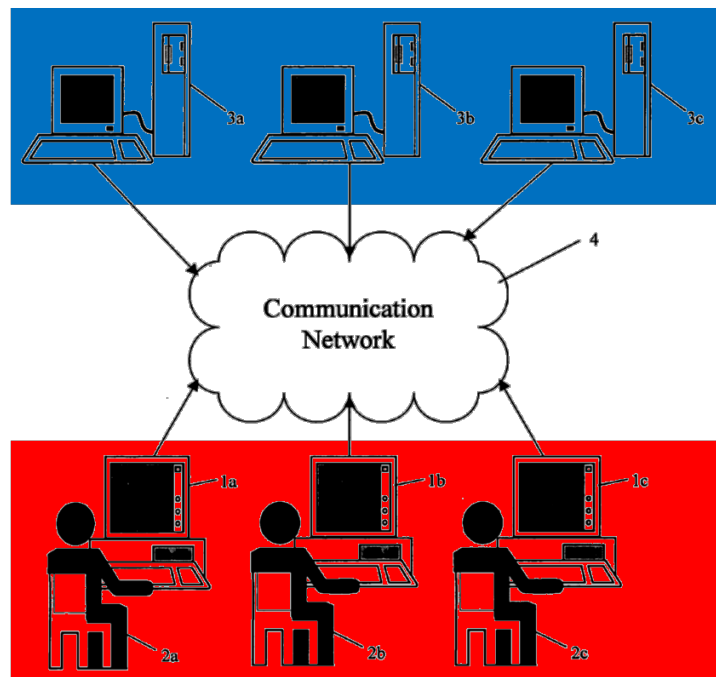


FIGURE 1

(EX1007, FIG.1; EX1002 ¶216.)

Moreover, *Abdolbaghian*, like *Wookey*, discloses a “keep-alive function” that “sends a ping, message or other signal to a first application to prevent the first application from being timed out.” (*Compare* EX1007, Abstract with EX1005 ¶[1136] (“client machine 10 *periodically* transmits a signal to the remote machine 30 to confirm that a connection is still intact”).) Accordingly, a POSITA would have had reason to consider the teachings of *Abdolbaghian* when implementing *Wookey*’s process (in context of the combined *Wookey-Eggert* system/process) given they

Petition for *Inter Partes* Review
Patent No. 10,375,215

disclose features in a common technical field and address similar problems. (EX1002 ¶217; *see also infra* Section IX.B.1.b.2.)

Abdolbaghian explains that the user terminal 1 and connected computer 3 communicate over a “communication network 4 [which] may include wire line ... or wireless [] connections.” (EX1007, 2:41–44; *see also id.*, 1:14–16, 2:36–40). Further, *Abdolbaghian* provides that “[i]nformation communicated over the communication network 4 may conform to any data communications protocol, including TCP/IP.” (*Id.*, 2:42–44; EX1002 ¶218.)

Abdolbaghian explains that in cases where the user terminal is operating an application that is, e.g., hosted by the connected computer, it is often undesirable to have that application timeout (e.g., terminated) based on an application timeout period. (EX1007, 1:14–36, 2:8–23, 2:49–65, 3:1–5, 4:9–11, FIGS. 1–3; EX1002 ¶219.) To solve this problem of avoiding an application timeout, *Abdolbaghian* provides a keep-alive mechanism that “sends a ping, message or other signal to a first application to prevent the first application from being timed out.” (EX1007, 1:63–67; *see also* EX1005 ¶[1136] (*Wookey* similarly disclosing that the above-noted periodically transmitted signals “to confirm that a connection is still intact” are designed in part to avoid an application session from being disconnected); EX1002 ¶220.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

Abdolbaghian discloses that the user terminal 1 sets a keep-alive function timeout clock for its keep-live function based on the longer application timeout period. (EX1007, 1:63–2:7, 4:25–37, FIG. 3.) For instance, *Abdolbaghian* explains, with reference to Figure 3, that “[b]ased on [timeout function] information, the keep-alive function may set its own function timeout clock (the ‘process clock’).” (*Id.*, 5:21–24; *see also id.*, Abstract (“The keep-alive function may, periodically or based on a timeout clock related to the first application, check the status of the first application and/or send a keep-alive input.”).) The “[keep-alive] function timeout clock may be synchronized with the application timeout clock.” (*Id.*, 5:24–25.) *Abdolbaghian* also provides that “[t]he value of the [keep-alive] function timeout clock may be set so that the [keep-alive] function timeout clock *will expire before* the application timeout clock expires.” (*Id.*, 5:25–27; *see also id.*, 5:43–46.) Thus, the keep-alive mechanism in *Abdolbaghian* has a clock or timer based on a keep-alive period that is separate from the application’s timeout clock or timer, but the keep-alive period is based on the application’s timeout period. (EX1002 ¶221; *see also id.* ¶¶222–223.)

Accordingly, *Abdolbaghian* discloses the concept of setting a keep-alive period that would have kept, e.g., an application on a connection alive, to be shorter

Petition for *Inter Partes* Review
Patent No. 10,375,215

than a longer timeout period based on which the application itself would timeout (e.g., due to disconnection). (EX1002 ¶224.)

(2) Reasons to Combine

A POSITA would have been motivated in light of *Abdolbaghian* to configure the code in the combined *Wookey-Eggert* method so that **client machine 10** implements the above-discussed keep-alive mechanism to send a keep-alive packet at a frequency set to *expire before* the abort timeout timer expires in the connection. (EX1002 ¶225.) Such an implementation would have been a predictable and straightforward combination of well-known technologies using known methods. *See KSR*, 550 U.S. at 416–18.

For example, a POSITA would have been motivated to implement *Abdolbaghian*'s teachings regarding setting a keep-alive period based on a longer timeout period (like the abort timeout value in the combined *Wookey-Eggert* process/system) because it would have allowed for the combined process/system to reset the abort timeout timer. (EX1002 ¶226.) A POSITA facing the wide range of needs created by developments in the technical field of *Wookey*, *Eggert*, and *Abdolbaghian*, would have appreciated the benefits of timing the communication of a keep-alive packet of the *Wookey-Eggert* combination based on a keep-alive period that is, in turn, based on the idle time period. (*Id.*)

Petition for *Inter Partes* Review
Patent No. 10,375,215

Indeed, the setting of a keep-alive period in a keep-alive mechanism to be less than a longer timeout value (such as the abort timeout value in the combined *Wookey-Eggert* process/system) for a connection was a well-known concept in the art before the '215 patent. (EX1002 ¶227; EX1012, 2:45–3:61, 11:43–46, 12:12–21, 15:2–26; EX1013, 5:48–63; EX1014, 4:38–43, 11:41–12:3.) Thus, setting the keep-alive period based on a longer timeout period would have been a common sense approach since the purpose of a keep-alive mechanism in the networking field is to keep something (e.g., the connection between two nodes) alive before the connection is subject to be terminated (e.g., based on a longer timeout value such as the abort timeout value). (EX1002 ¶228.)

Therefore, based on *Abdolbaghian*'s disclosure and knowledge in the art, the keep-alive time period in the combined *Wookey-Eggert* method would have been set to a time period that is *less than* any session's negotiated abort timeout value because to set the time period to a larger (or the same) value as the abort timeout value would render *Wookey*'s keep-alive mechanism impracticable. (EX1002 ¶229.)

Furthermore, setting of a keep-alive timer with a period less than the timeout value in the above-combination would have been a straightforward application of basic networking principles for maintaining a connection, as acknowledged by the '215 patent itself. (EX1002 ¶230; EX1001, 1:54–46, 2:10–16; EX1012, 2:45–3:61,

Petition for *Inter Partes* Review
Patent No. 10,375,215

11:43–46, 12:12–21, 15:2–26; EX1015, 10 (describing that a DeadTimer used to terminate communication between two nodes that “is 4 times the value of the Keepalive timer”); EX1013, 5:48–63; EX1014, 4:38–43, 11:41–12:3.)²² Such knowledge, coupled with the disclosures/suggestions of *Abdolbaghian*, would have motivated a POSITA to set the keep-alive timer in the combined system/process to a value less than the negotiated abort timeout value in order to prevent termination of the connection between **client machine 10** and **remote machine 30’**. (EX1002 ¶231.) Indeed, providing such an implementation to the *Wookey-Eggert* combination (where the keep-alive mechanism’s timer is set less than the negotiated abort timeout value) would have been a practical, common sense approach to ensure that *Wookey*’s modified keep-alive mechanisms are operational to facilitate intended network communications. (*Id.*)

²² Depending upon the operational objective of the system, a POSITA would have understood that there are various ways the combined system could have set the keep-alive period, including, e.g., setting the keep-alive period to a longer period than the abort timeout value (idle time period), which would have also resulted in the keep-alive period being based on the idle time period. (EX1002 ¶230 n.15; EX1016, 10.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

The combination of *Wookey* and *Eggert* with *Abdolbaghian*, similar to the *Wookey-Eggert* combination, would have involved the use of known technologies (e.g., aspects of similar protocols including the keep-alive mechanism) and design concepts and processes to obtain the foreseeable result of a reliable connection between *Wookey*'s client and server. (EX1002 ¶232.) Indeed, the above-modification would have involved a logical configuration to the code used by **client machine 10** where the keep-alive mechanism's timer is set less than the negotiated abort timeout value, as discussed above. (*Id.*) A POSITA would have been skilled and knowledgeable to configure *Wookey*'s system/method to implement a keep-alive timer set less than the negotiated abort timeout value, while taking into account any known programming, design, and other related concepts, limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. (*Id.* ¶233.) Thus, a POSITA would have had a reasonable expectation of success in the above-modification. (*Id.*) See *Pfizer, Inc.*, 480 F.3d at 1364.

Accordingly, the *Wookey-Eggert-Abdolbaghian* discloses and/or suggests the limitations of claim 4. (EX1002 ¶234.)

2. Claim 8

- a) **The method of claim 4 wherein the code, to which the access is provided, is configured such that, in response to being used by the client node, causes the client node to operate such that the keep-alive period is administered using a keep-alive timer.²³**

The *Wookey-Eggert-Abdolbaghian* discloses and/or suggests this limitation for the reasons below and those discussed above for claim 4. (*Supra* Section IX.B.1; EX1002 ¶¶235–237.)

As discussed for claim 4, the *Wookey-Eggert-Abdolbaghian* combination discloses and/or suggests “wherein the code, to which the access is provided, is configured such that, in response to being used by the client node, causes the client node to operate such that” *Wookey*’s keep-alive mechanism is configured to send one or more keep-alive packets based on a keep-alive function clock (keep-alive timer) that dictates the frequency for sending the keep-alive packets, where the keep-alive function clock is based on the abort timeout period (“idle time period”). (*Supra* Section IX.B.1.) Thus, the *Wookey-Eggert-Abdolbaghian* combination discloses and/or suggests that the keep-alive period is administered using a keep-alive timer. (*Id.*; EX1002 ¶¶236–237)

²³ See *supra* n.11.

3. Claim 9

- a) **The method of claim 8 wherein the code, to which the access is provided, is configured such that, in response to being used by the client node, causes the client node to operate such that the keep-alive timer is separate from an idle timer that is used to administer the idle time period.**²⁴

The *Wookey-Eggert-Abdolbaghian* discloses and/or suggests this limitation for the reasons below and those discussed above for claims 4 and 8. (*Supra* Sections IX.B.1–2; EX1002 ¶¶238–241.)

For example, as discussed above, the *Wookey-Eggert-Abdolbaghian* combination discloses that the client machine would have included a keep-alive function clock (“keep-alive timer”) that is set based on the negotiated ATO value (“idle time period”). (*Supra* Sections IX.B.1–2.) And as discussed above, the negotiated ATO value in the *Wookey-Eggert* combination is used to detect a time at which the connection between **client machine 10** and **remote machine 30’** is subject to deactivation. (*Supra* Sections IX.A.1.f–g.) *Eggert* explains that the abort timeout duration within the ATO field is used to track periods during which there is no communication to keep the connection active. (EX1006, 1–3; *see also id.*, 9 (“Long abort timeout values allow hosts to tolerate extended periods of

²⁴ *See supra* n.11.

Petition for *Inter Partes* Review
Patent No. 10,375,215

disconnection”); EX1002 ¶239.) Thus, in order to detect the expiration of the ATO value, the combined method/process would have implemented a timer based on the ATO value. Indeed, a POSITA would have known that separate timers could be configured and maintained to track separate time periods in networking applications such as those described in *Wookey*, *Eggert*, and *Abdolbaghian*. (EX1002 ¶240; EX1005 ¶¶[0864] (*Wookey* describing a timer for attempts to retransmit data packets), [0945] (describing another timer), [1093] (same), [1116] (discussing “multiple timers” to track certain events); EX1007, Abstract, 5:15–36 (discussing separate clocks for the keep-alive function and application timeout function); EX1015, 9–10; EX1012, Abstract, 5:66–6:21.) Thus, the *Wookey-Eggert-Abdolbaghian* combination discloses a keep-alive function clock (“keep-alive timer”) that is separate from another timer used to administer the abort timeout (“idle time period”) which is based on the ATO value. (EX1002 ¶241.)

X. EGGERT IS A PRINTED PUBLICATION

Eggert is an IETF Internet-Draft (“ID”) and is prior art under at least 35 U.S.C. §102(b) (pre-AIA) because it was published in April, 2004. The declaration of Alexa Morris, Managing Director of IETF, confirms *Eggert* was published, disseminated, and reasonably available to the public by April, 2004. (EX1019 ¶¶1-3, 9–10.)

IDs were at the time, and continue to be, working documents published through the IETF Secretariat and disseminated to the public by IETF through various media so that others may comment on them. (*Id.* ¶¶4–6.) Since 1998 (including 2004 and today), the IETF Secretariat publishes an ID on its public website, and publication announcements were sent to an IETF mailing list and relevant working group mailing lists. (*Id.* ¶¶5–6.) Anyone could have subscribed to IETF mailing lists, and the archives of all IETF mailing lists are publicly available on IETF’s website. (*Id.* ¶¶6–8.)

Eggert was published by the IETF in April, 2004. (*Id.* ¶¶9–10.) In 2004, a POSITA could have learned about *Eggert* in various ways, such as through announcements on the IETF announce mailing list, discussions on the IETF tcpm working group mailing list, or review of the archives of the IETF announce or tcpm mailing lists. (*Id.*)

Petition for *Inter Partes* Review
Patent No. 10,375,215

Collectively, this demonstrates that *Eggert* was published and publicly available in 2004 and at least prior to 2010. *Polycom, Inc., v. Directpacket Research, Inc.*, IPR2019-01235, Paper 19, 30–32 (Jan. 13, 2020) (finding a similar IETF ID was publicly available based on similar evidence).

XI. DISCRETIONARY DENIAL IS NOT APPROPRIATE

A. *Eggert* is Not Cumulative

Eggert is not cumulative to any prior art the Examiner considered during prosecution of the '215 patent and no arguments similar to those contained herein were ever presented to or considered by the Office. Nonetheless, PO may assert here (as it did in the Google -845 IPR) that RFC 5482 (EX1016) listed on the face of the '215 patent is substantially similar to *Eggert*. It is not, as the Board found in the Google -845 IPR. Google -845 IPR, Paper 16 at 14-17. The Board should reach the same conclusion here.

First, the Board routinely institutes trial where references in an IPR were considered in an IDS but not relied upon to reject claims during prosecution. *See, e.g., Amber.IO, Inc. D/B/A Two Tap v. 72Lux, Inc. D/B/A Shoppable*, IPR2020-00015, Paper 8 at 18-20 (April 1, 2020) (declining to exercise discretion where references were cited in an IDS but no evidence that the references were substantively considered) (citing *Advanced Bionics, LLC v. Med-El*

Petition for *Inter Partes* Review
Patent No. 10,375,215

Elektromedizinische Geräte GmbH, IPR2019-01469, Paper 6 (February 13, 2020) (precedential)); *Apple, Inc. v. Omni Medsci, Inc.*, IPR2020-00029, Paper 7 at 52–55 (April 22, 2020) (similar). Consistent with those proceedings, RFC 5482 was cited during prosecution along with twenty other references in an IDS with no explanation regarding any relevant disclosures. (EX1004, 70–72.) Moreover, RFC 5482 was never relied upon by the Examiner to reject the claims. (*Id.*) Even assuming *Eggert* is cumulative to RFC 5482, which it is not as explained below, *Eggert* was not considered in the light being presented herein (e.g., with supporting expert testimony and in combination with *Wookey*).

Second, as the Board previously found, *Eggert*'s protocol is fundamentally different from that discussed in RFC 5482 because, in contrast with RFC 5482, *Eggert*'s protocol calls for a common negotiated value for the timeout. Google -845 IPR, Paper 16 at 14–17. While *Eggert* discloses a negotiation protocol, RFC 5482 was purposefully designed such that “an exchange of UTO [user timeout] options between both ends of a connection is not a binding negotiation.” (EX1016, 4.)

Eggert also has meaningful disclosures that do not appear in RFC 5482 such as section 2.1 (including Figure 2). For example, it is *Eggert*'s mechanisms disclosed with reference to Figure 2 regarding exchange of ATO information during the three-way handshake that are relied upon to teach the claimed receive, detect,

Petition for *Inter Partes* Review
Patent No. 10,375,215

identify, and determine steps. (*Supra* Sections IX.A.1.d–g.) In contrast, RFC 5482 “does not define a mechanism to negotiate support of the TCP User Timeout Option [UTO] during the three-way handshake.” (EX1016, 9.) Therefore, these “material differences between the asserted art and the prior art involved during examination” weigh against discretionary denial. *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8, 17-18 (Dec. 15, 2017) (precedential as to §III.C.5, first paragraph); *see also* Google -845 IPR, Paper 16 at 16–17. Thus, *Eggert* is not cumulative to RFC 5482.

Accordingly, Petitioner requests that the Board institute review.

B. The Related Litigation Provides No Basis For Discretionary Denial

The Board should not exercise its discretion to deny institution under 35 U.S.C. § 314(a). *NHK Spring Co., Ltd. v. Intrix-Plex Techs, Inc.*, IPR2018-00752, Paper 8 (Sept. 12, 2018) does not apply here as no trial date has been scheduled in the related litigation. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 3 (Mar. 20, 2020) (precedential) (“*NHK* applies ... where the district court has set a trial date to occur earlier than the Board’s deadline to issue a final written decision in an instituted proceeding.”). The six-factor test addressed in *Fintiv* (“*Fintiv* factor(s)”) favors institution. *See id.*, 5–6.

Petition for *Inter Partes* Review
Patent No. 10,375,215

The **first *Fintiv* factor** (stay) is neutral, if not weighing in favor of institution. While there is currently no stay pending IPR resolution before the Western District of Texas (“WDTX”) court, Google intends to pursue such a stay should institution be granted. The Board should not speculate as to how the court would rule on the requested stay based on actions taken in different cases. *Western Digital Corp. et al. v. Martin Kuster*, IPR2020-01391, Paper 10 at 8-9 (February 16, 2020) (finding this factor neutral despite Patent Owner assertions regarding WDTX Judge Albright’s past practices regarding stays). Moreover, Google filed a motion to transfer in the district court litigation. The Court has currently stayed the litigation pending resolution of Google’s transfer motion. (EX1035.)

The **second *Fintiv* factor** (proximity of trial dates) weighs strongly in favor of institution. *First*, the district court has not set a trial date, which “weighs significantly against exercising [] discretion to deny institution of the proceeding.” *Google LLC v. Uniloc 2017 LLC*, IPR2020-00441, Paper 13 at 35 (July 17, 2020); *Nested Bean, Inc. v. Big Beings USA PTY LTD*, IPR2020-01234, Paper 15 at 14 (Jan. 25, 2021) (same). In fact, while the parties submitted proposed schedules, the court granted Google’s motion to stay the case pending resolution of Google’s transfer

Petition for *Inter Partes* Review
Patent No. 10,375,215

motion and declined to set a case schedule.²⁵ (EX1035; EX1029.) Accordingly, determining at this time whether the Board’s anticipated FWD date (e.g., Sept.-Oct. 2022) will occur after trial in the related litigation would be speculative.

Even if the parties’ most recent proposed schedule was considered as a baseline for comparison (which Google submits would be improper at this time), dates in that proposed schedule would need to be delayed because they were proposed prior to the court’s recent stay order and did not take the stay into consideration. (EX1035; EX1029, 5–9.) Accordingly, the proposed dates in light of the court’s stay order demonstrates that trial will likely occur after August 2022.²⁶ (EX1029, 6, 9.) Thus, any projected trial date (*albeit* speculative) would be more

²⁵ Disposition of Google’s transfer motion has taken priority over other activities. (EX1035.) *See also In re Apple Inc.*, 979 F.3d 1332, 1337 (Fed. Cir. 2020) (“disposing of a filed transfer motion ‘should unquestionably take top priority.’”).

²⁶ Consistent with the court’s practice, the proposed order stages deadlines and trial date based on the proposed August 2021 *Markman* hearing and that the “actual trial date” may materially differ from the schedule and that the court would consider “reasonable amendments to the case schedule post-*Markman*.” (EX1029, 9 n.3; *id.*, 6.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

likely after the expected due date of the Board’s FWD (e.g., around September-October 2022).

The Board has instituted trial even where an *actual* litigation trial date precedes the FWD deadline by two months or more. *Sand Revolution II, LLC v. Cont’l Intermodal Grp–Trucking LLC*, IPR2019-01393, Paper 24 at 8-10 (June 16, 2020) (“*Sand*”) (informative) (factor 2 favored institution despite WDTX trial preceding FWD deadline by five months); *Western Digital Corp.*, IPR2020-01391 at 9-10 (factor 2 found to be neutral despite WDTX trial predating FWD by three and a half months); *Apple Inc. v. Parus Holdings, Inc.*, IPR2020-00686, Paper 9 at 11–13, 22 (instituting trial where WDTX trial was two months before FWD deadline); *SMIC, Am. v. Innovation Foundry Techs., LLC*, IPR2020-00786, Paper 10, at 20–21 (Oct. 5, 2020) (same, where WDTX trial date preceded FWD by one month); *Fintiv*, IPR2020-00019, Paper 11 at 5, 9 (“an early trial date” is “non-dispositive” and simply means that “the decision whether to institute will likely implicate other factors”).

It is also unlikely that trial in WDTX will proceed without delay. WDTX has issued a suspension order every month for the past eleven months suspending almost all trials in the district from March 13, 2020 to at least March 31, 2021 due to the COVID-19 pandemic, creating a large backlog of trials. (*See* EX1030; *see also*

Petition for *Inter Partes* Review
Patent No. 10,375,215

EX1031, 2 (agreeing to stipulation to postpone trial date “to allow the COVID-19 situation to ameliorate”); EX1034 (showing J. Albright’s docket including 792 cases filed in 2020).) Thus, as the Board has recognized, civil trials in WDTX “may possibly slip ... [due to] months of backlogged trials, including many active criminal cases that would take precedence over civil trials.” *HP Inc. v. Slingshot Printing LLC*, IPR2020-01085, Paper 12 at 7 (Jan. 14, 2021); *Sand* at 8-10.

Finally, the potential for transfer based on Google’s transfer motion further reduces the likelihood that trial will precede the Board’s FWD. *Dish Network, L.L.C. v. Broadband iTV, Inc.*, IPR2020-01267, Paper 15 at 17-18 (Jan. 21, 2021) (explaining that the second *Fintiv* factor “is a proxy for the *likelihood* that the trial court will reach a decision on validity issues before the Board reaches a [FWD]”). Indeed, transfer of the WDTX case to the proper venue (the Northern District of California) will undoubtedly delay any trial beyond the anticipated FWD due date. *Cf. Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 8, at 7 n.7 (Mar. 27, 2020) (denying institution based in part on scheduled trial date that was later vacated upon transfer); *see Uniloc 2017 LLC v. Google LLC*, Case No. 18-CV-00502, Dkt No. 277 (E.D. Tex. June 19, 2020) (transfer order vacating unreached deadlines).

The **third *Fintiv* factor** (investment in parallel proceedings) also weighs in favor of institution. The case is in earliest stages, and as discussed, all deadlines

Petition for *Inter Partes* Review
Patent No. 10,375,215

other than those concerning Google's transfer motion have been stayed. Thus, investment by the parties in invalidity and claim constructions issues will be delayed, especially if the case is transferred. And even if the transfer motion is denied, using the above-discussed proposed schedule as an exemplary baseline (which will require adjustments), any *Markman* hearing in the case is likely not to occur until August 2021 or later, and thus would occur near the expected due date of the Board's institution decision. (EX1029, 6.) Discovery will not commence, and dispositive motions will not be due, until after the *Markman* hearing. Other case activity that has taken place or will eventually take place (outside claim construction) that does not relate to validity should "not weigh in [the Board's] consideration of this issue." *Western Digital Corp.*, IPR2020-01391 at 11. Such minimal investment is insufficient to support discretionary denial. *Juniper Networks, Inc. v. Huawei Digital Techs. (Cheng Du) Co., Ltd.*, IPR2020-01130, Paper 13 at 12-13 (Jan. 22, 2021) (factor 3 weighs against discretionary denial when claim construction order has been entered, but fact and expert discovery has not closed, and dispositive motions and briefings have not been submitted); *Dish* at 19-21 (similar); *HP* at 7 (similar). Additionally, Google's diligence in filing this Petition just three months

Petition for *Inter Partes* Review
Patent No. 10,375,215

after receiving PO's narrowed list of asserted claims²⁷ further weighs against discretionary denial. *Dish* at 20-21 (petitioner's diligence in filing the petition weighed against discretionary denial); *Fintiv* at 11.

The **fourth *Fintiv* factor** (overlap) is neutral if not in favor of institution. There is, at this early stage of the district court litigation, no evidence of overlap with the grounds here and any invalidity positions to be pursued in the district court. Indeed, preliminary invalidity contentions have not yet been served. And, as discussed above, the district court must first resolve the pending transfer motion before addressing substantive issues, including invalidity. Given the stay of the litigation, it is likely expert discovery relating to invalidity will not be completed until at the earliest mid-2022 and dispositive motions on those issues would be due

²⁷ PO initially asserted 455 claims across eight patents (including 34 claims from the '215 patent). (EX1017, 1.) The WDTX court ordered PO to substantially reduce the asserted claims to approximately 65 claims. On October 20, 2020, PO narrowed its asserted claims as ordered, but "reserve[d] the right to either narrow the claims further, **substitute claims**, or **add a reasonable number of claims**." (EX1032.) On December 4, 2020, PO informed Petitioner that it will not modify its narrowed list of asserted claims "**at this time**." (EX1033.)

Petition for *Inter Partes* Review
Patent No. 10,375,215

after that. Accordingly, substantive invalidity issues would not be addressed in the litigation until well past the institution decision and likely very near the time of the Board's expected FWD. Moreover, Petitioner may rely on invalidity grounds not asserted here in the district court litigation, including system art and prior art references and combinations not raised herein. Therefore, any potential overlap between this proceeding and the district court is minimal and speculative at this time.

Other circumstances (*Fintiv* factor six) also favor institution. The Petition presents strong grounds demonstrating the unpatentability of the challenged claims, which weighs against discretionary denial. (*See supra* Section IX.) *Western Digital Corp.*, IPR2020-01391 at 14-15. Indeed, the Board has already instituted the Google -845 IPR and the Unified -742 IPR challenging claims of the related '995 patent. (*See supra* Section II.) The Google -845 IPR is based in part on *Eggert*, which is being applied here. Moreover, this petition is the only challenge to the '215 patent before the Board, which is a "crucial fact" favoring institution. *Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 10 at 6 (May 12, 2020).²⁸

²⁸ Discretionary denial under *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper No. 19 is not applicable here given the '215 patent is not at issue in any other proceeding pending before the Board.

Petition for *Inter Partes* Review
Patent No. 10,375,215

While ***Fintiv* factor 5** (same parties) may weigh slightly in favor of denial, the remaining factors are at least neutral, if not strongly favoring institution. Further, even if the Board finds that **factor 1** (or even **factor 2**) somehow tilts in favor of denial, the investment of resources, lack of evidence of overlap, and/or strength of Petitioner’s unpatentability positions (**factors 3, 4, 6**) outweigh these other factors. Accordingly, based on a “holistic view” of whether integrity of the system and efficiency is best served, institution here is proper. *Samsung Elecs. Co. Ltd. v. Dynamics Inc.*, IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

Petition for *Inter Partes* Review
Patent No. 10,375,215

XII. CONCLUSION

For the reasons given above, Petitioner requests institution of IPR for claims 1, 4, 8, and 9 of the '215 patent based on each of the grounds specified in this petition.

Respectfully submitted,

Dated: March 15, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,375,215

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,375,215 contains, as measured by the word-processing system used to prepare this paper, 13,989 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: March 15, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,375,215

CERTIFICATE OF SERVICE

I hereby certify that on March 15, 2021, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,375,215 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on PAIR:

Patrick Caldwell (Reg. No. 44580)
G. Gordon (Reg. No. 64517)
P.O. Box 59655
Dept. SVIPGP
Dallas, TX 75229

A courtesy copy was also sent via electronic mail to the Patent Owner's litigation counsel at the following addresses:

TD-PTAB@devlinlawfirm.com
ddahlgren@devlinlawfirm.com
dlflitparas@devlinlawfirm.com
1526 Gilpin Ave.
Wilmington, DE 19806

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)

EXHIBIT 7

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

JENAM TECH, LLC,
Patent Owner

Patent No. 10,069,945

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,069,945**

Petition for *Inter Partes* Review
Patent No. 10,069,945

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	1
III.	PAYMENT OF FEES	3
IV.	GROUND FOR STANDING.....	3
V.	PRECISE RELIEF REQUESTED	3
A.	Claims for Which Review Is Requested	3
B.	Statutory Ground of Challenge	3
VI.	LEVEL OF ORDINARY SKILL	4
VII.	OVERVIEW OF THE '945 PATENT	5
VIII.	CLAIM CONSTRUCTION	8
IX.	DETAILED EXPLANATION OF GROUNDS.....	10
A.	Ground 1: The Combination of <i>Wookey</i> and <i>Berg</i> Render Obvious Claims 1, 9–10, 34, and 64–69	10
1.	Claim 1	10
2.	Claim 9	52
3.	Claim 10	53
4.	Claim 34	57
5.	Claim 64	68
6.	Claim 65	70
7.	Claim 66	71
8.	Claim 67	71
9.	Claim 68	72

Petition for *Inter Partes* Review
Patent No. 10,069,945

10.	Claim 69.....	72
X.	DISCRETIONARY DENIAL IS NOT APPROPRIATE	73
XI.	CONCLUSION.....	81

Petition for *Inter Partes* Review
Patent No. 10,069,945

LIST OF EXHIBITS

EX1001	U.S. Patent No. 10,069,945
EX1002	Declaration of Bill Lin, Ph.D.
EX1003	Curriculum Vitae of Bill Lin, Ph.D.
EX1004	Prosecution History of U.S. Patent No. 10,069,945
EX1005	U.S. Pre-Grant Publication No. 2007/0171921 to Wookey <i>et al.</i>
EX1006	RESERVED
EX1007	U.S. Patent No. 6,674,713 to Berg <i>et al.</i>
EX1008	DARPA RFC 793 TRANSMISSION CONTROL PROTOCOL
EX1009	U.S. Patent No. 6,584,546 to Kavipurapu
EX1010	U.S. Patent No. 9,923,995
EX1011	Bova <i>et al.</i> , RELIABLE UDP PROTOCOL <draft-ietf-sigtran-reliable-udp-00.txt> 25 February 1999
EX1012	U.S. Pre-Grant Publication No. 2004/0093376 to De Boor <i>et al.</i>
EX1013	U.S. Patent No. 7,535,913 to Minami <i>et al.</i>
EX1014	U.S. Pre-Grant Publication No. 2005/0204013 to Raghunath <i>et al.</i>
EX1015	U.S. Pre-Grant Publication No. 2007/0005804 to Rideout
EX1016	U.S. Pre-Grant Publication No. 2004/0098748 to Bo <i>et al.</i>
EX1017	<i>Jenam Tech, LLC's Second Amended Set of Infringement Contentions</i> regarding U.S. Patent No. 10,069,945 (March 17, 2021)
EX10181	IETF RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1
EX1019	U.S. Patent No. 7,636,805 to Rosenberg
EX1020	U.S. Patent No. 6,212,175 to Harsch

Petition for *Inter Partes* Review
Patent No. 10,069,945

EX1021	U.S. Patent No. 8,259,716 to Diab
EX1022	U.S. Patent No. 6,665,727 to Hayden
EX1023	U.S. Patent No. 6,981,048 to Abdolbaghian <i>et al.</i>
EX1024	<i>Jenam Tech., LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Joint Submission of Proposed Amended Scheduling Order, ECF No. 57 (W.D. Tex. Mar. 4, 2021)
EX1025	Fourteenth Supplemental Order Regarding Court Operations Under the Exigent Circumstances Created By the COVID-19 Pandemic, The United States District Court for the Western District of Texas (March 17, 2021)
EX1026	<i>Digital Retail Apps, Inc. v. H-E-B, LP</i> , Case No. 6:19-cv-00167, Joint Stipulation and Order Postponing Trial, ECF No. 182 (W.D. Tex. Jan. 13, 2021)
EX1027	<i>Jenam Tech, LLC's</i> Preliminary Narrowing of Claims (October 20, 2020)
EX1028	<i>Jenam Tech, LLC's</i> Correspondence Regarding Narrowing of Claims (December 4, 2020)
EX1029	Judge Alan D. Albright's Case Statistics By Year (Retrieved from DocketNavigator on March 9, 2021)
EX1030	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Order Granting Motion to Stay Case, ECF No. 58 (W.D. Tex. Mar. 10, 2021)
EX1031	RESERVED
EX1032	U.S. Patent No. 7,464,326 to Kawai
EX1033	IETF RFC 1866 Hypertext Markup Language – 2.0
EX1034	RESERVED
EX1035	<i>Jenam Tech, LLC's First Set of Infringement Contentions</i> regarding U.S. Patent No. 10,069,945 (August 21, 2020)

Petition for *Inter Partes* Review
Patent No. 10,069,945

I. INTRODUCTION

Google LLC (“Petitioner”) requests *inter partes* review (“IPR”) of claims 1, 9–10, 34, and 64–69 (“the challenged claims”) of U.S. Patent No. 10,069,945 (“the ’945 patent”) (EX1001) assigned to Jenam Tech, LLC (“Patent Owner” or “PO”). For the reasons below, the challenged claims should be found unpatentable and canceled.

II. MANDATORY NOTICES

Real Parties-in-Interest: Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies the following as the real parties-in-interest: Google LLC.¹

Related Matters: The ’945 patent is asserted in the following civil actions: *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.); and *Jenam Tech, LLC v. Samsung Group*, Case No. 4:20-cv-00279 (E.D. Tex.) (voluntarily dismissed).

The ’945 patent claims priority to U.S. Patent Application No. 12/714,454 filed February 27, 2010. (EX1001, Cover.) U.S. Patent No. 9,923,995 (“the ’995

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

Petition for *Inter Partes* Review
Patent No. 10,069,945

patent”) also claims priority to U.S. Patent Application No. 12/714,454. (EX1010, Cover.) The ’995 patent is subject to the following instituted IPRs: *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845 (“Google -845 IPR”); and *Unified Patents, LLC v. Jenam Tech, LLC*, IPR2020-00742 (“Unified -742 IPR”).

Petitioner is also concurrently filing another IPR petition challenging the ’945 patent and IPR petitions challenging U.S. Patent No. 10,306,026 which is also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.).

Petitioner has also filed the following IPR petitions challenging patents related to the ’995 and ’945 patents, and which are also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.):

- IPR2021-00627 (U.S. Patent No. 10,375,215, “the ’215 patent”);
- IPR2021-00628 (U.S. Patent No. 10,075,564, “the ’564 patent”);
- IPR2021-00629 (the ’564 patent); and
- IPR2021-00630 (U.S. Patent No. 10,075,565, “the ’565 patent”).

Counsel and Service Information: Lead counsel: Naveen Modi (Reg. No. 46,224). Backup counsel: (1) Joseph E. Palys (Reg. No. 46,508), (2) Quadeer A. Ahmed (Reg. No. 60,835), (3) Jason Heidemann (Reg. No. 77,880).

Petition for *Inter Partes* Review
Patent No. 10,069,945

Service information is Paul Hastings LLP, 2050 M Street NW, Washington, DC 20036, Tel.: 202.551.1700, Fax: 202.551.1705, email: PH-Google-Jenam-IPR@paulhastings.com. Petitioner consents to electronic service.

III. PAYMENT OF FEES

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-2613.

IV. GROUNDS FOR STANDING

Petitioner certifies that the '945 patent is available for review, and Petitioner is not barred/estopped from requesting review on the grounds herein.

V. PRECISE RELIEF REQUESTED

A. Claims for Which Review Is Requested

Petitioner requests review and cancellation of claims 1, 9–10, 34, and 64–69 as unpatentable based on the following ground.

B. Statutory Ground of Challenge

Ground 1: Claims 1, 9–10, 34, and 64–69 are unpatentable under pre-AIA 35 U.S.C. § 103(a) over U.S. Publication No. 2007/0171921 to Wookey *et al.* (“Wookey”) (EX1005) in view of U.S. Patent No. 6,674,713 to Berg *et al.* (“Berg”) (EX1007).

The '945 patent issued from an application filed March 7, 2018, which claims priority through a number of applications back to an application filed February 27,

Petition for *Inter Partes* Review
Patent No. 10,069,945

2010. (EX1001, 1:8–28.) Petitioner assumes for this proceeding only, without conceding, that the earliest effective filing date of the '945 patent is February 27, 2010.

Wookey was published on July 26, 2007, from an application filed on November 14, 2006. (EX1005, Cover.) *Berg* was issued on January 6, 2004, from an application filed on February 23, 1999. (EX1007, Cover.) Thus, *Wookey* and *Berg* qualify as prior art at least under 35 U.S.C. § 102(b) (pre-AIA).

The above references were not considered during prosecution of the '945 patent (EX1001, Cover (“References Cited”); *see also generally* EX1004) and are not the same or substantially like any art previously presented to the Office.

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art at the time of the alleged invention of the '945 patent (“POSITA”) would have had an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking. (EX1002 ¶¶16–18.)² More education can supplement practical experience and vice versa. (*Id.*)

² Petitioner submits the declaration of Dr. Bill Lin (EX1002), an expert in the field of the '945 patent. (EX1002 ¶¶3–15; EX1003.)

VII. OVERVIEW OF THE '945 PATENT

The '945 patent is directed to networking, and to the sharing of information for detecting an idle TCP connection. (EX1001, 2:26–28, 8:33–55; EX1002 ¶¶64–72.) Figure 7 illustrates such a process.

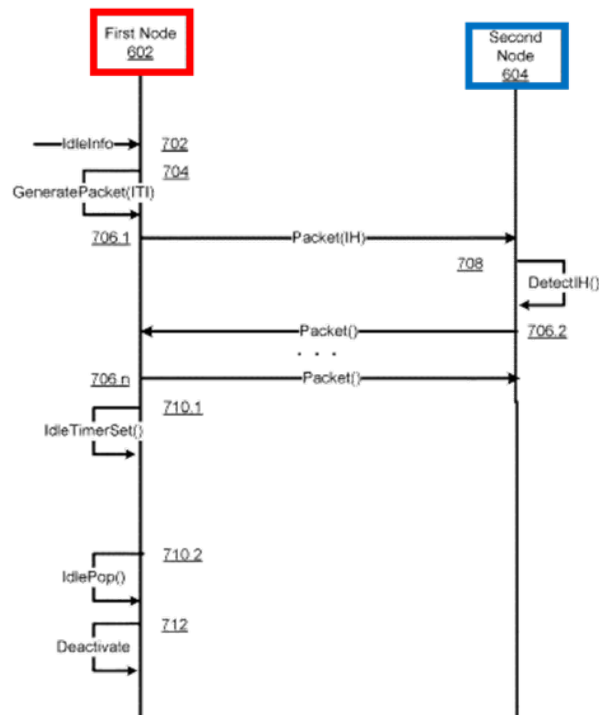
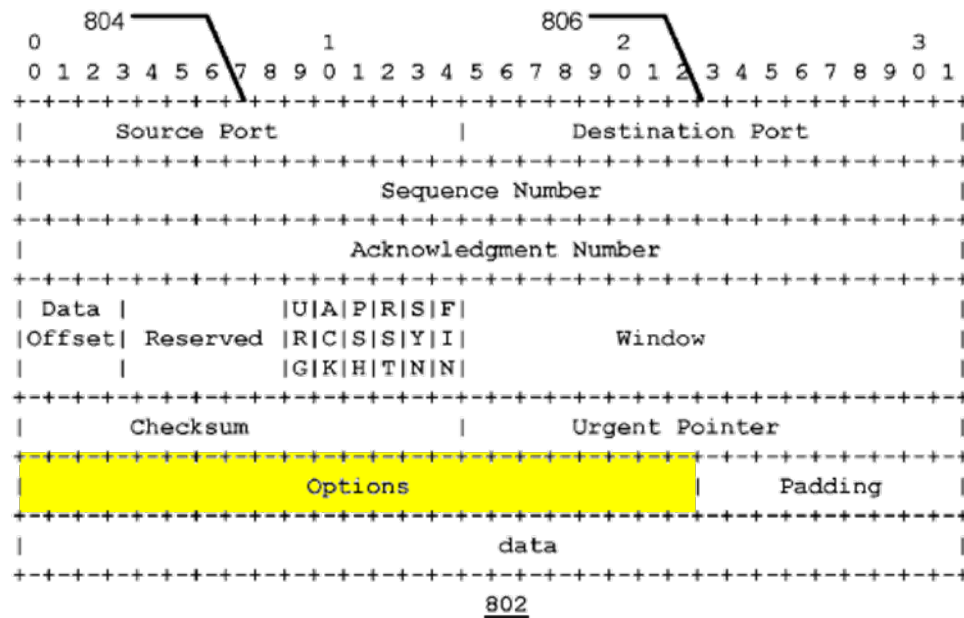


Fig. 7

(EX1001, FIG. 7 (annotated); EX1002 ¶64.)

First, the **first node 602** receives a message 702 that identifies idle information representing a duration for an idle time period (“ITP”). (EX1001, 11:59–67.) Message 702 may take various forms, such as “a message received via

a network.” (*Id.*, 11:62–67.) The idle information “may include and/or identify a duration of time” for detecting an ITP. (*Id.*, 12:40–44.) The duration “may be specified according to various measures of time[,] including seconds.” (*Id.*; EX1002 ¶65.)



Petition for *Inter Partes* Review
 Patent No. 10,069,945

The TCP options field contains a KIND sub-field that identifies the type of option presented, a length sub-field that specifies the length of the option field, and an ITP Data sub-field containing data. (EX1001, FIG. 8.)

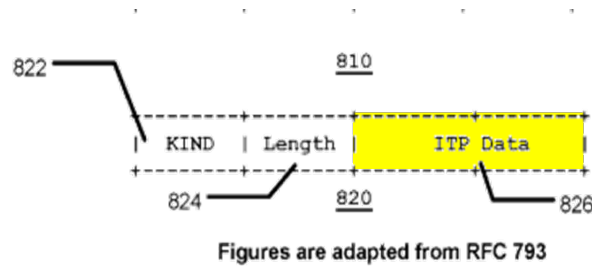


Fig. 8

(*Id.* (cropped-annotated); EX1002 ¶66.)

The ITP header is exchanged during the three-way handshake (EX1001, 14:53–67.) For example, the **first node** transmits a message 706.1, i.e., a TCP packet including an ITP header (IH) containing ITP information, to the **second node 604**. (*Id.*, 16:31–33, FIG. 7.) Message 708 exemplifies the **second node**'s detection of the ITP header in the received TCP packet. (*Id.*, 21:52–55; EX1002 ¶67; *see also* EX1002 ¶¶68–71.)

All the limitations in the challenged claims were known in the prior art and obvious. (*See* Section IX; EX1002 ¶72; *see also* EX1002 ¶¶19–63 (technology background), citing EXS. 1008–1009, 1011–1016, 1018–1023.)

VIII. CLAIM CONSTRUCTION

Under the applicable standard in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), claim terms are typically given their ordinary and customary meanings as understood by a POSITA at the time of the invention based on the claim language, specification, and the prosecution history of record. *Phillips*, 415 F.3d at 1313; *see also id.* at 1312–16. The Board, however, only construes the claims when necessary to resolve the controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper No. 11 at 16 (Aug. 14, 2015) (citation omitted). Petitioner believes no express constructions of any claim terms are necessary to assess whether the prior art reads on the challenged claims. (EX1002 ¶73.)

Claims 1 and 34 recite the term “a second protocol that is separate from the TCP.” (EX1001, 24:38–39, 27:33–34.) The ’945 patent mentions “separate from the TCP” only in the “Summary” section, which the ’945 patent explains is “not an extensive overview of the disclosure and it does not identify key/critical elements of the invention or delineate the scope of the invention.” (*Id.*, 2:32–39, 2:63, 3:28–29.) As discussed below, *Berg*’s RUDP protocol is separate from the TCP protocol. (*See, e.g., infra* Section IX.A.1.f.)

Moreover, the Board preliminarily found *Berg* discloses a non-TCP protocol in the Unified -742 IPR. *See, e.g.,* Unified -742 IPR, Paper 11 (Institution Decision)

Petition for *Inter Partes* Review
Patent No. 10,069,945

at 10–11. Thus, because Petitioner relies on *Berg*’s similar disclosures to satisfy the “protocol that is separate from the TCP” terms herein (*see infra* Sections IX.A.1.f, IX.A.4.f.), and given the ’945 patent offers no evidence requiring a special meaning of this term and the prior art discloses this feature under any reasonable interpretation, construction of this term is unnecessary.³

³ Petitioner reserves all rights to raise claim construction and other arguments, including challenges under 35 U.S.C. §§ 101 or 112, in district court as relevant to those proceedings. *See, e.g., Target Corp. v. Proxicom Wireless, LLC*, IPR2020-00904, Paper 11 at 11–13 (November 10, 2020). A comparison of the claims to any accused products in litigation may raise controversies that are not presented here given the similarities between the references and the patent.

IX. DETAILED EXPLANATION OF GROUNDS**A. Ground 1: The Combination of *Wookey* and *Berg* Render Obvious Claims 1, 9–10, 34, and 64–69****1. Claim 1****a) A computer-implemented method, comprising:**

To the extent that the preamble of claim 1 is limiting, *Wookey* discloses the limitations therein. (EX1002 ¶¶98–105.)

Wookey discloses a computer-implemented method, as claimed. For example, *Wookey* teaches “a *method* for making a hypermedium page interactive,” using a client machine and one or more remote machines. (EX1005, Abstract; *id.* ¶[0002]; EX1002 ¶100; *see also* EX1002 ¶¶74–83 (*Wookey* overview).)⁴

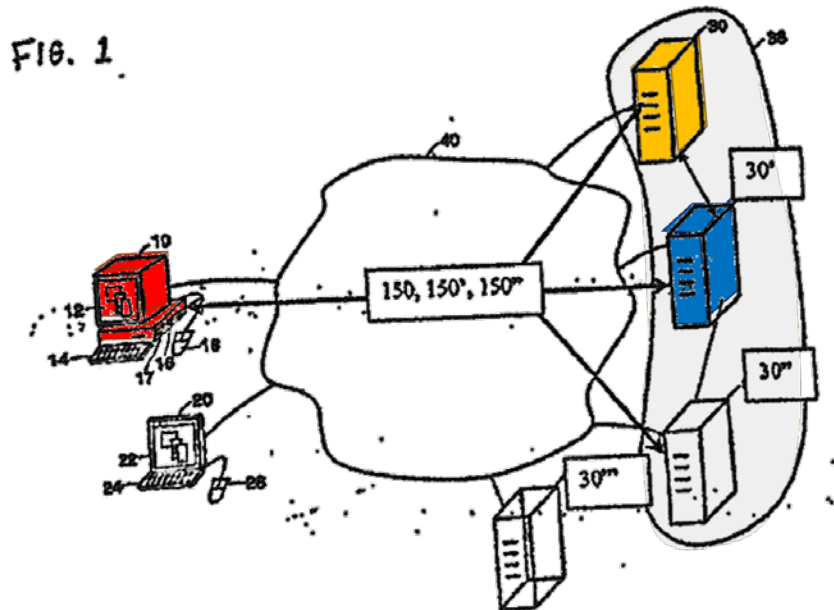
Figures 1, 2C, and 3D below show an exemplary environment in which a **client machine 10**, **remote machine 30**, and **remote machine 30'** are employed to carry out the method.⁵ (*See also* EX1005 ¶¶[0020], [0024], [0026], [0192]–[0196],

⁴ Emphasis is added unless otherwise stated.

⁵ *Wookey*'s disclosures (including its figures) include certain typographical errors when referencing its client machines (referring to them interchangeably as 10, 10' or 20). (EX1002 ¶104; EX1005 ¶¶[0135]–[0136], [0152]–[0154], [0192]–[0196], [0207]–[0216], FIGS. 1, 2C, 3D.) For example, the bottom figure in Fig. 2C

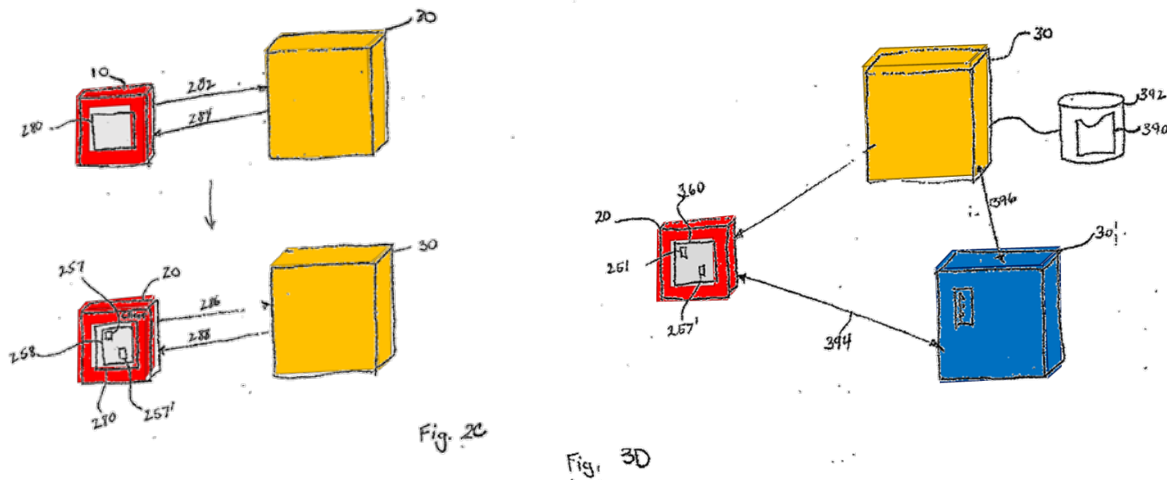
Petition for *Inter Partes* Review
 Patent No. 10,069,945

[0207].) *Wookey*'s client and remote machines are "typical computers" (*id.* ¶[0021], FIGS. 1A–1B) including personal computers and wireless devices (e.g., mobile telephones or other portable telecommunication devices) (*id.* ¶¶[0171]–[0173]).



erroneously shows client machine "20," but the corresponding disclosure for both figures in Fig. 2C describes client machine "10." (EX1005 ¶¶[0192]–[0196].) Moreover, in context of *Wookey*'s disclosures, a POSITA would have understood that any teachings regarding one client machine 10/10'/20 in *Wookey* apply equally to the other client machines. (EX1002 ¶104.)

Petition for *Inter Partes* Review
 Patent No. 10,069,945



(*Id.*, FIGS. 1, 2C, 3D (each annotated); EX1002 ¶¶101–103.)

Thus, *Wookey* discloses “[a] computer-implemented method,” as claimed.

(*See infra* Sections IX.A.1.b–j; EX1002 ¶105.)

b) causing access to be provided to a server computer including:

- (1) a non-transitory memory storing a network application, and**
- (2) one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application to operate in accordance with a first protocol including a transmission control protocol (TCP);**

Wookey discloses this limitation. (EX1002 ¶¶106–123.) For example, *Wookey* discloses that its above-noted computer-implemented method causes access to be provided to a **remote machine 30** (“server computer”). (EX1005 ¶[0160] (“the

Petition for *Inter Partes* Review
 Patent No. 10,069,945

remote machines 30 ... are provided as ... computer servers”). The access is caused to be provided, for example, by allowing **client machine 10** to access **remote machine 30** to gain access to resources provided by **remote machine 30**, as discussed further below with respect to claim limitations 1.d–f. (*See infra* Sections IX.A.1.d–f.)

Remote machine 30 is configured with typical computer architecture, as shown in Figures 1A–1B below (EX1005 ¶¶[0016], [0021].)

FIG. 1A

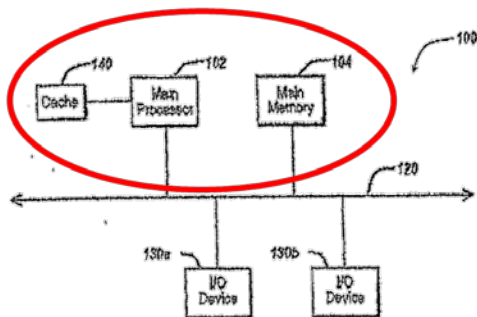
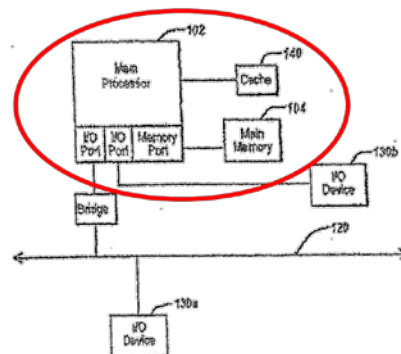


FIG. 1B



(*Id.*, FIGS. 1A and 1B (annotated); EX1002 ¶108.)

In this architecture, **remote machine 30** includes a main memory unit 104 and cache memory 140 (individually or collectively, the claimed “non-transitory memory”) which are in communication (e.g., via a system bus 120) with a main processor 102 (“one or more processors”). (EX1005 ¶¶[0161]–[0165]; EX1002 ¶108.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

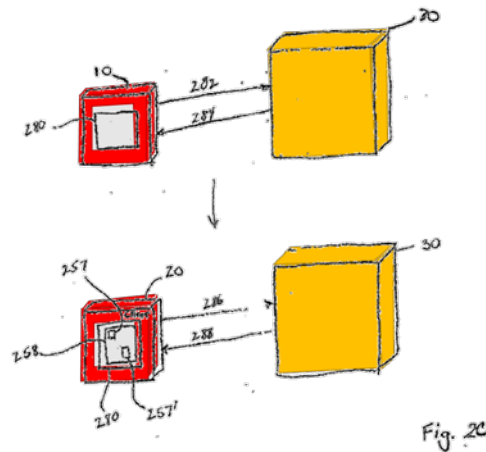
Main memory 104 and cache memory 140 store instructions (e.g., software). For instance, “*instructions [are] fetched from the main memory unit 104*” by main processor 102. (EX1005 ¶¶[0162]–[0163]; *id.* ¶[0158], FIGS. 1A–1B.)⁶ Also, processor 102 communicates with cache memory 140, which stores data and instructions accessible to the processor. (*Id.* ¶[0165]; EX1002 ¶¶109–111 (*citing* EX1009, 1:15–39).)

Main memory 104 and cache memory 140 store a network application, as claimed, because the above instructions (e.g., software) enable **remote machine 30** to act as a web server (“network application”). A web server is an example of a network application because it includes software that runs on one computer (e.g., **remote machine 30**) which provides communication to another application (e.g., web browser 280) running on another computer (e.g., **client machine 10**). (EX1005 ¶¶[0179], [0207], [0213].) *Wookey* describes an arrangement in Figure 2C where **remote machine 30** “acts as a web server” at least because it communicates with

⁶ *Wookey* interchangeably uses “main processor 102,” “microprocessor 102,” “processor 102,” and “central processing unit 102”. (EX1002 ¶109 n.5; EX1005 ¶¶[0161] (“a central processing unit 102”), [0163] (“microprocessor 102”), [0164] (“processor 102”), FIGS. 1A–B.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

client machine 10 to authenticate, and provide a list of resources to, **client machine 10**. (*Id.* ¶¶[0207], [0213].)



(*Id.*, FIG. 2C (annotated); EX1002 ¶112.)

Main processor 102 executes instructions for executing **remote machine 30** as a web server, such that it operates in accordance with a first protocol including a transmission control protocol (TCP), as claimed. For example, **client machine 10** executes a web browser application 280 (EX1005 ¶[0192]) that “transmits a request 282 to access a Uniform Resource Locator (URL) address corresponding to an HTML [HyperText Markup Language] page residing on” **remote machine 30** (*id.* ¶[0193]). (*Id.* ¶¶[0024], [0207], [0213], [0279].) The request is received and handled by **remote machine 30**. (*E.g., id.* ¶¶[0193]–[0196].) **Remote machine 30** includes functionality for authenticating (*id.* ¶[0194]) and transmitting an HTML

Petition for *Inter Partes* Review
Patent No. 10,069,945

page to **client machine 10** (*id.* ¶[0196]). (*See also id.* ¶¶[0179], [0207], [0213]–[0215], [0803]–[0808] (web server functionality), FIGS. 2C, 3D; EX1002 ¶¶113–117.)

The communication link 150 between **client machine 10** and **remote machine 30** uses the TCP. (EX1005 ¶¶[0154]–[0156], [0174], [0731]–[0732].) Thus, in the communication between **client machine 10** and **remote machine 30** for Figure 2C, **remote machine 30** acts as a web server that operates in accordance with a first protocol including TCP, as claimed. (EX1002 ¶¶118–120.)

Accordingly, *Wookey's* **remote machine 30**, which can perform processes over a TCP connection to handle requests from **client machine 10**, executes instructions via main processor 102 such that its web server operates in accordance with a first protocol including a TCP. (*Id.* ¶122.) Given processor 102 performs the processing functionalities of **remote machine 30** (EX1005 ¶[0021], FIGS. 1A–1B), processor 102 would thus execute the instructions such that **remote machine 30** acts as a web server in accordance with a first protocol including a TCP, to process, e.g., the received request 282 from **client machine 10**. (EX1002 ¶¶121, 123.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

- c) causing a TCP connection to be established with a client computer, by:
 - communicating a segment including at least one first synchronize bit;
 - communicating a first acknowledgement of the segment, and at least one second synchronize bit; and
 - communicating a second acknowledgement;

Wookey discloses this limitation. (EX1002 ¶¶124–129.) As discussed, **remote machine 30** (“server computer”) communicates with **client machine 10** (“client computer”)⁷ over a TCP connection. (*Supra* Section IX.A.1.b.) As explained below, when the machines are communicating over a TCP connection, *Wookey*’s method would have necessarily caused a TCP connection to be established between **remote machine 30** and **client machine 10** by performing a three-way handshake, which would involve the three “communicating” processes recited in this limitation. (EX1002 ¶124.)

A three-way handshake method is necessarily performed to establish a TCP connection. (*Id.* ¶125; *see also id.* ¶¶46–47.) For example, RFC 793 (EX1008), the

⁷ Figures 1A–1B depict block diagrams of typical computer architectures for both remote machines and client machines (e.g., **client machine 10**). (EX1005 ¶[0021].) Thus, **client machine 10** is a computer for similar reasons discussed above. (*Supra* Section IX.A.1.b; EX1002 ¶124 n.7.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

TCP specification, provides that a three-way handshake is a necessary procedure used to establish a TCP connection. (EX1008, 32, 34; *id.*, 31, 34–37.) The three-way handshake was known to involve transmitting three messages to negotiate and start a TCP-based session between two devices. (EX1002 ¶126; EX1008, 31–32; 34–37.) The ’945 patent acknowledges such understandings, citing RFC 793. (EX1001, 14:53–67.)

The *first* communicated message (step 1) includes a segment with a control bit for synchronization (SYN bit) and thus discloses “communicat[ing] a segment including at least one first synchronize bit,” as claimed. (EX1008, 31; EX1001, 14:53–61.) The *second* communicated message (steps 2 and 3 combined) includes an acknowledgment (ACK) to the segment with the SYN bit and a second SYN bit, and thus discloses “communicat[ing] a first acknowledgement of the segment, and at least one second synchronize bit,” as claimed. (EX1008, 31–32; EX1001, 14:62–63, 14:65–67.) The *third* communicated message (step 4) includes a second ACK to the second SYN bit and thus discloses “communicat[ing] a second acknowledgement,” as claimed. (EX1008, 31–32; EX1001, 14:64–67 EX1002 ¶127.)

Thus, to establish a TCP connection with **client machine 10**, **remote machine 30** necessarily performs a three-way TCP handshake that includes the same features

Petition for *Inter Partes* Review
Patent No. 10,069,945

recited in this limitation because this handshake was/is a *required* procedure for establishing a TCP connection. (EX1002 ¶¶128–129.) Indeed, PO relies on the same three-way handshake disclosures in RFC 793 for its allegations regarding this limitation. (EX1017, 5–6.)⁸

d) **causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer;**⁹

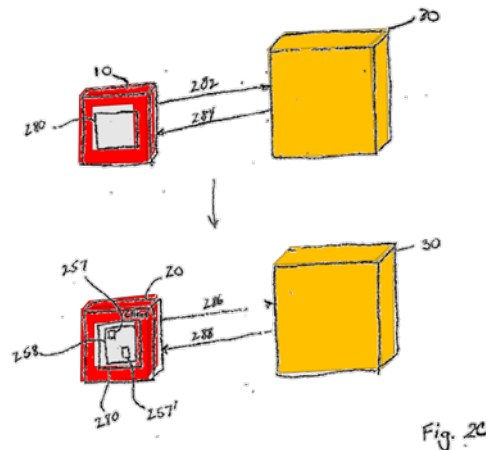
Wookey discloses this limitation. (EX1002 ¶¶130–136.) For example, *Wookey* discloses that upon **client machine 10** transmitting a request 282 to **remote machine 30** for accessing a URL address corresponding to an HTML page residing on **remote machine 30**, **remote machine 30** is caused to communicate an HTML

⁸ Petitioner’s references to PO’s infringement allegations are not indicative of any concession that any accused instrumentality infringes any claim limitation as alleged by PO.

⁹ The ’945 patent specification does not describe the claimed “causing first data ...” features as recited in the challenged claims. (EX1002 n.9.) Therefore, while Petitioner addresses this limitation under this Ground, Petitioner does not concede that the ’945 patent provides adequate disclosures of such features.

Petition for *Inter Partes* Review
 Patent No. 10,069,945

page 284 which is an authentication page (“first data”) to **client machine 10** utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user the **client machine 10**. (EX1005 ¶¶[0192]–[0194], FIG. 2C.)



(*Id.*, FIG. 2C (annotated); EX1002 130.)

In particular, **client machine 10** executes a web browser application 280 that “transmits a request 282 to access a Uniform Resource Locator (URL) address corresponding to an HTML page residing on remote machine [30]”. (EX1005 ¶¶[0192]–[0193].) **Remote machine 30** is then caused to communicate an HTML page 284 to **client machine 10**. (*Id.* ¶[0193]; EX1002 ¶131.)

HTML page 284 is “an authentication page [284] that seeks to identify the client machine 10 or the user of the client machine 10” (EX1005 ¶[0193]) and

Petition for *Inter Partes* Review
Patent No. 10,069,945

includes data such as tags and associated information defining the layout of a webpage and what is to be presented on the webpage. (EX1002 ¶132; *see also* EX1005 ¶¶[0196], [0207]–[0214] (discussing an exemplary webpage that **remote machine 30** transmits to **client machine 10**), [0754]–[0755] (same).)

The understanding that an HTML page including data meets this limitation is consistent with PO’s interpretation of this limitation. (EX1017, 6 (alleging “causing first data to be communicated from the server computer to the client computer (e.g., device that receives the HTML pages, etc.).”).)

“The authentication page [284] allows the client machine 10 to transmit user credentials, via the web browser 280, to the remote machine 30 for authentication.” (EX1005 ¶[0194], FIG. 2C.) Thus, the HTML authentication page 284 is “for being presented to a user of the client computer,” as claimed, because the **client machine 10**’s user has to review the authentication request presented on web browser 280 and provide credentials in response. (*Id.*; *see also id.* ¶¶[0178]–[0179]; EX1002 ¶133.)

Moreover, the HTML authentication page 284 is communicated from **remote machine 30** to **client machine 10** “utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP),” as claimed. As discussed above, the communication link 150 between **client machine 10** and **remote machine 30** is a TCP connection operating in accordance with the TCP

Petition for *Inter Partes* Review
Patent No. 10,069,945

protocol. (*See supra* Sections IX.A.1.b–c.) *Wookey* further discloses that this communication link 150 uses HTTP at the application layer. (EX1005 ¶[0155] (“[t]he communications link 150 may use ... any application layer protocol, such as the Hypertext Transfer Protocol (HTTP)”); *see also id.*, ¶¶[0159], [0551]–[0556] (describing that the server agent and client agent can communicate using the HTTP protocol), [0899].) Thus, *Wookey*’s method causes **remote machine 30** to act as a web server to communicate HTML pages to **client machine 10** over a TCP connection in accordance with TCP and HTTP, for being presented to the user of **client machine 10**. (EX1002 ¶¶134–136.)

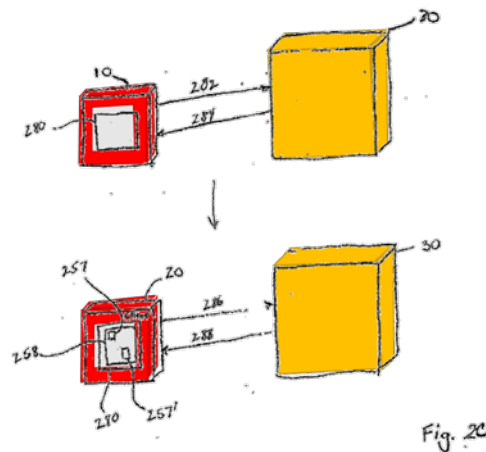
Petition for *Inter Partes* Review
Patent No. 10,069,945

e) causing the server computer to permit second data, from the user of the client computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP);¹⁰ and

Wookey discloses this limitation. (EX1002 ¶¶137–140.) For example, *Wookey*’s method causes **remote machine 30** to permit user credentials (“second data”), from the user of **client machine 10**, to be received at **remote machine 30** from **client machine 10** utilizing the TCP connection in accordance with TCP and HTTP. (EX1005 ¶¶[0192]–[0196], FIG. 2C.)

¹⁰ The ’945 patent specification does not describe the claimed “causing ... to permit second data ...” features as recited in the challenged claims. (EX1002 n.11.) Therefore, while Petitioner addresses this limitation under this Ground, Petitioner does not concede that the ’945 patent provides adequate disclosures of such claimed features.

Petition for *Inter Partes* Review
 Patent No. 10,069,945



(*Id.*, FIG. 2C (annotated); EX1002 ¶137.)

As discussed above, the communication link 150 between **client machine 10** and **remote machine 30** is a TCP connection operating in accordance with the TCP protocol and HTTP. (*See supra* Sections IX.A.1.b–d.) In response to receiving the HTML authentication page 284, **client machine 10** transmits user credentials (e.g., via communication 286) to **remote machine 30** for authentication over this TCP connection. (*Id.*; EX1005 ¶¶[0193]–[0196], FIG. 2C; EX1002 ¶138.)

In particular, *Wookey* discloses that “[t]he authentication page allows the client machine 10 to transmit user credentials, via the web browser 280, to the remote machine 30 for authentication” and these “[t]ransmitted user credentials are verified” by one of the remote machines. (EX1005 ¶[0194].) *Wookey* similarly discloses that “an access control decision is made based on *received information* about the user

Petition for *Inter Partes* Review
Patent No. 10,069,945

resources available to the user of the client system,” where “[t]he remote machine 30 may verify the *user credentials received* from the client machine 10” or “the remote machine 30 may pass the *user credentials* to another remote machine for authentication.” (*Id.* ¶[0195].) Thus, *Wookey*’s method causes **remote machine 30** to permit (e.g., allow) the user credentials, from the user of **client machine 10**, to be received at **remote machine 30** because **remote machine 30** sends HTML authentication page 284 requesting such credentials from the user. (*Id.* ¶¶[0192]–[0196], FIG. 2C; EX1002 ¶¶139–140.) The received information regarding the user’s credentials is “data,” as claimed, because it is used to review and make decisions regarding the level of access control to resources that can be provided to **client machine 10**.

Petition for *Inter Partes* Review
Patent No. 10,069,945

f) causing access to be provided, to the client computer, to code that causes the client computer to operate in accordance with a second protocol that is separate from the TCP, in order to establish a second protocol connection with another server computer, by:¹¹

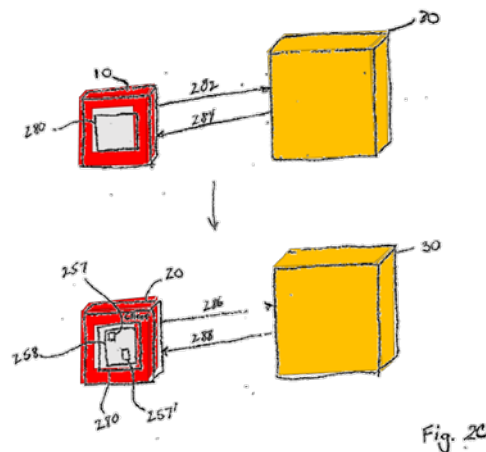
The *Wookey-Berg* combination discloses and/or suggests this limitation. (EX1002 ¶¶141–180.) For example, as discussed below, *Wookey*’s method causes access to be provided, to **client machine 10**, to an HTML page 288 containing encoded URLs, in order to establish a second protocol connection with **remote machine 30’** (“another server computer”) that is different from **remote machine 30**. As discussed below, it would have been obvious in light of *Berg*’s teachings to configure *Wookey*’s HTML page 288 containing the encoded URLs such that the HTML page 288 containing the encoded URLs would cause **client machine 10** to operate in accordance with an RUDP protocol (“second protocol that is separate from the TCP”), in order to establish an RUDP connection (“second protocol connection”) with **remote machine 30’**. (*Id.*)

For example, referring to Figure 2C below, once **client machine 10** is authenticated (e.g., based on the provided user credentials), “remote machine [30]

¹¹ The ’945 patent specification does not describe the claimed “code” features as recited in the challenged claims. (EX1002 n.12.) Therefore, while Petitioner

Petition for *Inter Partes* Review
 Patent No. 10,069,945

prepares and transmits to the client machine 10 an HTML page 288 that includes a Resource Neighborhood window 258 in which appears graphical icons 257, 257' representing resources to which the client machine 10 has access.” (EX1005 ¶[0196].)



(*Id.*, FIG. 2C (annotated); EX1002 ¶143.)

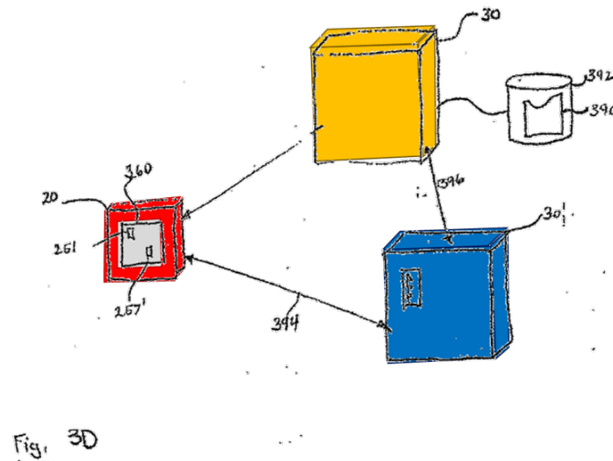
The HTML page 288 includes encoded URLs associated with icons (e.g., 257, 257') on the HTML page 288, and both (1) the encoded URLs included in the HTML page or (2) the HTML page including the encoded URLs disclose the claimed “code.” (EX1002 ¶144.) For example, “[e]ach icon 257, 257' is associated with *an*

addresses this limitation under this Ground, Petitioner does not concede that the '945 patent provides adequate disclosures of such claimed features.

encoded URL that specifies: the location of the resource ...; *a launch command associated with the resource*; and *a template identifying how the results of accessing the resource should be displayed.*” (EX1005 ¶[0216].) Each encoded URL also “*contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (*Id.*) Thus, when a user *clicks* an icon 257, 257’ corresponding to a resource from the displayed HTML page 288, **client machine 10** uses the encoded URL associated with the clicked icon to initiate the process for establishing a second connection (e.g., connection 394 in Figure 3D) with **remote machine 30**.¹² (EX1002 ¶144; *see also* EX1005 ¶¶[0026], [0213]–[0216], FIG. 3D.) Accordingly, the encoded URLs are the claimed “code” because such encoded URLs are instructions that, when used by **client machine 10**, command **client machine 10** to access resources from one or more locations (e.g., one or more remote machines). (EX1002 ¶145.)

¹² **Remote machine 30**’ is a computer for similar reasons discussed above regarding **remote machine 30** and is a server because it provides access to requested resources to **client machine 10** as explained below. (*Supra* Section IX.A.1.b; EX1005 ¶[0021]; EX1002 ¶144 n.13.)

Petition for *Inter Partes* Review
 Patent No. 10,069,945



(EX1005, FIG. 3D (annotated); EX1002 ¶145.)

The above understanding is consistent with how the '945 patent claims describe its “code” as an “instruction for connecting to the another server.” (EX1001, cl. 41 (28:20-22).) *Wookey*’s encoded URLs similarly include an “instruction,” such as the launch command to connect to **remote machine 30’**. (EX1002 ¶146.)

Moreover, the entire HTML page 288 (including the encoded URLs and icons) also discloses the claimed “code that causes the client computer,” because an HTML page was known to comprise HTML code used to define, e.g., what is displayed on the screen of the client node. (EX1002 ¶147 (citing EX1012).) The HTML code used to display information relating to page 288 would include the icons associated with the encoded URLs, and when **client machine 10** uses one of the

Petition for *Inter Partes* Review
Patent No. 10,069,945

encoded URLs, it causes **client machine 10** to operate in accordance with a second protocol separate from the TCP to set up a second protocol connection with another server (e.g., **remote machine 30'**). PO similarly relates the claimed code to “HTML pages” in its infringement assertions for this and similar limitations. (EX1017, 8 (“causing access to be provided, to the client computer (e.g., device that receives the HTML pages, etc.), to code.”).)

Wookey’s method further discloses “causing access to be provided,” to **client machine 10**, to this “code” because it discloses **remote machine 30** causing access to be provided, to **client machine 10**, to HTML page 288 including the encoded URLs, so **client machine 10** can request access to a resource represented by the icons 257/257’ on the HTML page 288. (EX1005 ¶[0196]; EX1002 ¶148.)

Thus, *Wookey* discloses, “causing access to be provided, to the client computer, to code that causes the client computer to operate” such that a process for establishing a second connection (e.g., connection 394 in Figure 3D) with **remote machine 30'** is initiated.¹³ (EX1002 ¶149.)

¹³ Petitioner’s analysis regarding the “code” limitation equally applies whether the “code” is the encoded URLs in the HTML page or the HTML page itself (including the encoded URLs). (EX1002 ¶149 n.15.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

Regarding the connection established between machines **10** and **30'**, **client machine 10** includes an “HTTP client agent,” such as the web browser application 280, which “can use any type of protocol.” (EX1005 ¶[0159]; *see also id.* ¶¶[0155], [0192]–[0195].) For example, connection 394 can be made using various protocols, including the Virtual Network Computing (VNC) protocol. (*Id.* ¶[0216].) VNC is a desktop-sharing system that allows a computer to remotely access and control another computer. (EX1002 ¶150; EX1014 ¶[0019].) VNC can run over various “industry standard transport protocols, such as TCP/IP, IPX/SPX, NetBEUI.” (EX1005 ¶[0225]; *see also id.* ¶¶[0155], [0215]–[0216].) IPX/SPX and NetBEUI protocols, being non-TCP–based transport protocols, are examples of protocols that are separate from the TCP protocol. (EX1002 ¶¶151–152; EX1013, 1:20–33.)

Wookey further explains that **client machine 10** may use “a different type of protocol than the one used to send the request to the remote machine 30.” (EX1005 ¶[0732]; *see also id.*, (“the client machine 10 may make a request to the remote machine 30 using the IPX protocol and request the address of the remote machine 30' as a TCP/IP protocol address”); EX1002 ¶153.)

Wookey also identifies desired characteristics associated at least with the second connection that a POSITA would have considered when choosing a protocol, such as negotiating connection parameters, reducing unintentional session

Petition for *Inter Partes* Review
Patent No. 10,069,945

terminations due to an imperfect connection, detecting/handling disconnections like when a mobile device enters an elevator, and ability to specify an inactive time before connection termination. (EX1002 ¶154; EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153].)

Thus, **client machine 10** identifies negotiable parameters for the connection to be established. (EX1002 ¶155.) For example, **client machine 10** and **remote machine 30'** “exchange a set of messages which negotiate the parameters under which communications will occur.” (EX1005 ¶[0774]; *see also id.* ¶¶[0738]–[0739] (describing the **remote machine 30'** listening for connection requests and processing them), [0744], [0772]–[0773].)

So when **client machine 10** uses one of the encoded URLs (e.g., the one pointing to **remote machine 30'**) associated with one of the icons on the HTML page 288, it results in **client machine 10** utilizing, e.g., the web browser application 280, to establish a connection (e.g., connection 394) with **remote machine 30'** in accordance with a transport protocol. (EX1002 ¶¶156–157.)

While *Wookey* discloses various protocols (including protocols that are separate from TCP (e.g., IPX)) may be used to establish the second connection and the desired characteristics of such a connection (EX1005, ¶[0225]), *Wookey* does not expressly provide details regarding establishing and maintaining the second

Petition for *Inter Partes* Review
Patent No. 10,069,945

connection or the types of negotiable parameters used with those protocols for the second connection. (EX1002 ¶158.) However, it would have been obvious to a POSITA to consider and implement a protocol following *Wookey*'s desired characteristics (e.g., such as those identified above) for the second connection between **client machine 10** and **remote machine 30'**, which may be "a different type of protocol than the one used to send the request to the remote machine 30." (EX1005 ¶[0732].)

Berg describes such a protocol (a "*protocol that is separate from the TCP*") that allows nodes to negotiate or modify timeout timers on a per-connection basis to provide a reliable and efficient network connection. Accordingly, based on the teachings of *Berg*, a POSITA's knowledge, and guidance from *Wookey*, it would have been obvious to configure *Wookey*'s **client machine 10** such that when the "code" (e.g., the encoded URL(s) in HTML page 288 pointing to a resource on **remote machine 30'**, or the HTML page 288 including such URL(s)) is used, the code causes **client machine 10** (e.g., by way of browser application 280) to operate per a second protocol such as the one described by *Berg* (which would have been separate from the TCP protocol used to send the request to **remote machine 30**) having at least some characteristics contemplated by *Wookey* for such a second

connection between **client machine 10** and **remote machine 30**'. (EX1002 ¶¶159–160.)

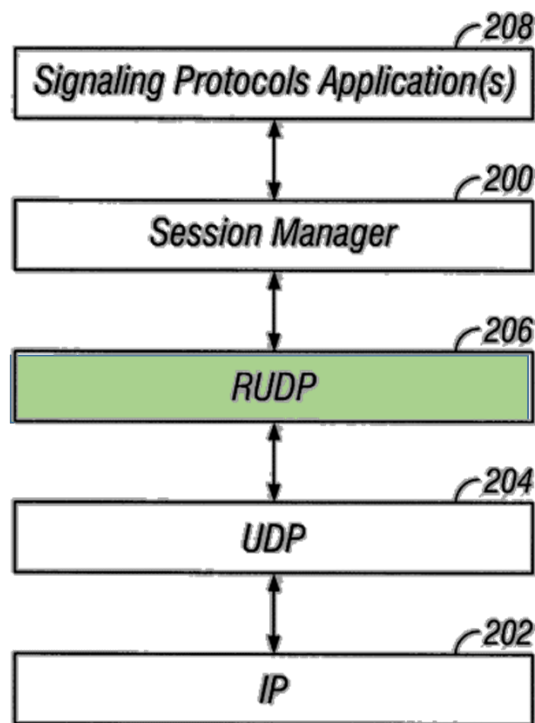
i. Berg

Berg, like *Wookey*, relates to establishing a reliable connection between a client and server as it states “[t]o provide a reliable backhaul, it is important to be able to maintain multiple IP connections or sessions between the signaling terminal device and the call processing device, so that message[s] can be transmitted even when the *inherently unreliable IP network fails*” and “[t]he system must be configured so that *communications are not interrupted upon network failure*, and so that *communications can resume when the network comes back up*.” (EX1007, 2:4–11; EX1002 ¶¶84–97 (*Berg* overview).) Thus, *Berg* discloses networking features in the same technical field as *Wookey*. (See, e.g., EX1005 ¶¶[0581] (*Wookey* disclosing that traversal of network segments by **client machine 10** may cause changes in its network address or host name or causes it “to disconnect”), [1135]–[1136] (discussing the need for connections to manage unintentional disconnection between the client and remote machines), [1153] (**client machine 10** losing connection because the user enters an elevator); *id.* ¶¶[0013], [0196], FIGS. 1, 2C, 3D.) A POSITA, therefore, would have had reason to consider *Berg*’s teachings when contemplating and implementing *Wookey*’s method for establishing a second

Petition for *Inter Partes* Review
Patent No. 10,069,945

protocol connection between **remote machine 30'** and **client machine 10**. (EX1002 ¶¶161–162.)

Berg provides a gateway device (acting as a client) connected to a media gateway controller (acting as a server). (EX1007, 3:14–23, 2:55–60.) In *Berg*'s system, both the client and server devices execute a session manager that manages the data network communication session. (*Id.*, 2:54–60, 7:63–8:2, 8:24–31; EX1002 ¶163.) The session manager “operates logically above a *reliable communication layer*”, which “determines when or if a session is connected or failed.” (EX1007, 2:54–60.) *Berg* explains “[a] protocol layer is ‘reliable’ when it guarantees that it will deliver or generate an error message for every message that an application program requests the transport layer using that protocol to transport.” (*Id.*, 8:27–31; *see also id.*, 8:11–13.) *Berg*'s Figure 2 below describes the layers that computer systems use for network communications based on the Open Systems Interconnection (OSI) reference model. (*Id.*, 8:3–17, 5:27–55.)

**FIG. 2**

(*Id.*, FIG. 2 (annotated); EX1002 ¶163.)

The **Reliable User Datagram Protocol (RUDP) 206** runs on top of the User Datagram Protocol (UDP) protocol software 204 at a transport layer. (EX1007, 8:10–17, 16:66–17–17; EX1002 ¶164.) The UDP is a non-TCP communications protocol (e.g., a protocol that is separate from the TCP) that is primarily used for establishing low-latency and loss-tolerating connections, especially time-sensitive transmissions, between applications. (EX1002 ¶164; EX1015 ¶[0110]; EX1016 ¶[0010].) The RUDP layer 206 allows *Berg*’s system to determine “when or if a

Petition for *Inter Partes* Review
Patent No. 10,069,945

session is connected or failed.” (EX1007, 2:54–60.)¹⁴ Above the **RUDP layer 206** is the Session Manager 200 and Signaling Software Application(s) 208. (*Id.*, 8:24–27.)

The Session Manager 200 can run above any reliable communication mechanism (e.g., RUDP 206). (*Id.*) A communication layer “is ‘reliable’ when it guarantees that it will deliver or generate an error message for every message that an application program requests the transport layer using that protocol to transport.” (*Id.*, 8:27–31; *see also id.*, 8:11–13; EX1002 ¶165.)

Berg notes that the “TCP/IP has a number of characteristics that make it unsuitable for” some applications. (EX1007, 17:28–30.) In particular, “[m]ost TCP/IP implementations that allow properties like timers to be modified, *do not allow* the modification to be done on a per-connection basis.” (*Id.*, 17:40–42; EX1002 ¶166.)

Contrasted to TCP/IP implementations, “RUDP is designed *to allow characteristics of each connection to be individually configured* so that many

¹⁴ *Berg* uses “RUDP layer 206” and “communication layer” to refer to the same layer in the exemplary OSI model in Figure 2. (EX1002 ¶164 n.18; EX1007, 2:54–60, 8:13, 8:24–28.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

protocols with different transport requirements can be implemented simultaneously on the same platform.” (EX1007, 17:42–46.) For example, RUDP allows the nodes to negotiate various parameters including the timeout values associated with timers for the retransmission timeout (*id.*, 20:24–29, 22:44–56), acknowledgement timeout (*id.*, 20:29–35, 23:15–24), null segment (*id.*, 20:36–41, 23:45–59), and transfer state (*id.*, 20:42–47, 24:15–25). *Berg* explains that the messages sent to set up the RUDP connection are UDP packets that include an RUDP header. (*Id.*, 17:59–18:20, 19:1–40; EX1002 ¶167.)

The “RUDP is a lightweight protocol layer designed to run on top of UDP [that] can provide reliable in-order delivery” (EX1007, 16:66–67; *see also id.*, 17:9–16) and it “has a *very flexible design* that would make it *suitable for a variety of transport uses*. (*Id.*, 17:13–15; EX1002 ¶168.)

Thus, the RUDP is “a second protocol that is separate from the TCP” (e.g., a non-TCP protocol), and operating in accordance with the RUDP would have caused a client to set up an RUDP connection (“a second protocol connection”) with a server. (EX1002 ¶169.) *See also* Unified -742 IPR, Paper 11 at 10 (Board agreeing that *Berg*’s RUDP is a non-TCP protocol).

ii. Reasons to Combine

Based on *Wookey*'s and *Berg*'s disclosures and a POSITA's knowledge at the time, it would have been obvious to configure and implement *Wookey*'s **client machine 10** (including its web browser application 280) to use a protocol like *Berg*'s RUDP ("a second protocol that is separate from the TCP") in order to set up an RUDP ("second protocol") connection with **remote machine 30'** (*see, e.g.*, EX1005, Figure 3D) in relation to claim limitations 1.f–j and claims 9–10. (EX1002 ¶170.) Such an implementation would have had many advantages and been a predictable and straightforward result achieved by combining well-known technologies using known methods. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 416–18 (2007).

Both *Wookey* and *Berg* are directed to establishing a reliable connection between two nodes, where one or both nodes may experience an unintentional disconnection, and thus disclose features in a similar technological field. (EX1002 ¶170; *supra* Section IX.A.1.f.i.) Thus, a POSITA would have had reason to consider *Berg* when contemplating and implementing the teachings of *Wookey*. (EX1002 ¶170.) And when collectively considered, such a POSITA would have been motivated to modify *Wookey*'s **client machine 10** to use a protocol like *Berg*'s RUDP in order to set up an RUDP connection with **remote machine 30'**. (*Id.* ¶171.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

As discussed, *Wookey* describes using “code” to cause **client machine 10**, e.g., a web browser application 280, to use a protocol that can be separate from the one used to connect **client machine 10** and **remote machine 30**, in order to set up a second protocol connection between **client machine 10** and **remote machine 30’**. *Wookey* also describes the types of characteristics concerning the protocol that can be used for the second connection, such as negotiating connection parameters, reducing unintentional terminations, detecting/handling disconnections, specifying a permissible inactive time before connection termination. (EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153]; *supra* Section IX.A.1.f.) A POSITA would have been thus motivated by *Wookey*’s disclosures/suggestions to consider and incorporate a protocol that would provide the desired features to *Wookey* for the second protocol connection with **remote machine 30’**, such as the RUDP described by *Berg*. (EX1002 ¶172.)

Using the RUDP protocol described in *Berg* with *Wookey*’s system/method would have provided **client machine 10** with an improved reliable connection to **remote machine 30’** with the versatility of negotiated timeout parameters consistent with *Wookey*’s desired characteristics for the second connection. *See KSR*, 550 U.S. 416–17. Indeed, by using a protocol like *Berg*’s RUDP protocol, *Wookey*’s method would enable negotiation of properties like the timeout values on a per-connection

Petition for *Inter Partes* Review
Patent No. 10,069,945

basis over a reliable connection (EX1007, 24:40–47), and provide alerts to the nodes if the session fails (*id.*, 2:55–60). *Wookey*’s method for “providing a client with a reliable connection to a host service” (EX1005 ¶[0077]) would therefore have been improved by providing a connection that is “very flexible ... [and] suitable for a variety of transport uses,” as provided by *Berg* (EX1007, 17:9–16). (EX1002 ¶173; EX1005 ¶[0903].) Accordingly, such a combination would have yielded the predictable result of providing a network communication exchange that minimizes disruptions caused by timeout issues. (EX1002 ¶176.) *See KSR*, 550 U.S. at 417.

It would have been a foreseeable and straightforward implementation to configure *Wookey*’s **client machine 10** such that the “code” (e.g., the encoded URLs in HTML page 288 or the HTML page 288 including such URLs), when used, would cause **client machine 10** to operate in accordance with a second protocol separate from the TCP like *Berg*’s RUDP. (EX1002 ¶174.) *See KSR*, 550 U.S. at 416. For example, as noted above, “*the URL includes a file, or a reference to a file, that contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (EX1005 ¶[0216].) In light of the above, a POSITA would have been motivated to configure *Wookey*’s encoded URL such that it includes information causing the client to establish a connection with **remote machine 30’** using a second protocol separate from the TCP (like *Berg*’s

Petition for *Inter Partes* Review
Patent No. 10,069,945

RUDP). (EX1002 ¶175.) This implementation would have been achieved by combining well-known technologies and methods, such as known network design concepts and technologies described above by *Wookey* and *Berg*, and known in the art at the time, to obtain the foreseeable result of a reliable connection between *Wookey*'s **client machine 10** and **remote machine 30'**. (*Id.* ¶175.)

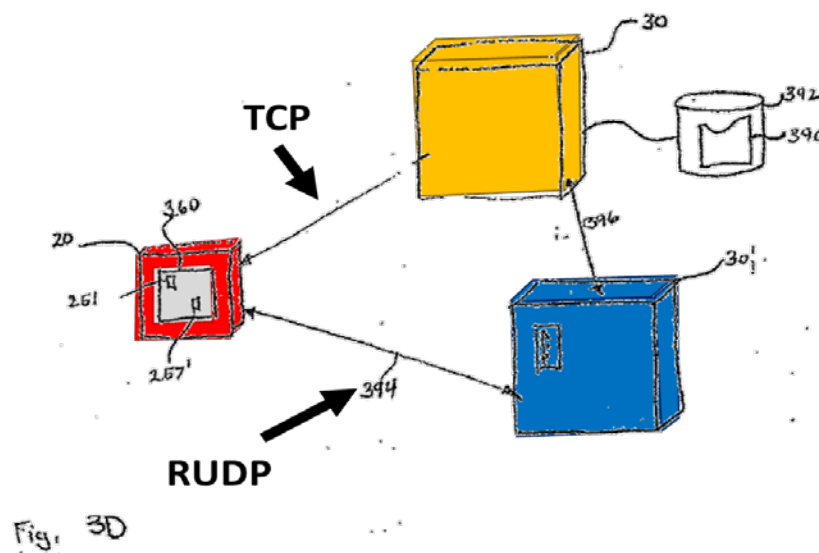
Indeed, the above-modification would have involved the substitution of features from one reliable protocol from a finite number of available alternative reliable communication protocols, such as that described by *Berg*. (*Id.* ¶177.) Thus, a POSITA would have had a reasonable expectation of success in the above-modification. (*Id.*) See *Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1364 (Fed. Cir. 2007) (“only a reasonable expectation of success, not a guarantee, is needed” in an obviousness analysis).¹⁵

A POSITA would have been skilled and knowledgeable to configure the above modification in various ways, while taking into account any known

¹⁵ There is no evidence of nonobviousness at this time that would overcome the overwhelming evidence of obviousness as presented herein. Petitioner reserves the right to address any such evidence/argument should PO later present any in this proceeding.

Petition for *Inter Partes* Review
 Patent No. 10,069,945

programming, design, and other related concepts, limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. (EX1002 ¶178.) For example, a POSITA would have been motivated based on such disclosures to configure *Wookey*'s system and process consistent with the non-limiting example reflected below.



(EX1005, FIG. 3D (annotated); EX1002 ¶178.)

* * *

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 1.f. (*Id.* ¶¶179–180; *see also infra* Sections IX.A.1.g–j.)

g) receiving a packet,

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶181–187.) As explained below, the “code” in the combined *Wookey-Berg* system/method would, when used by **client machine 10**, result in **client machine 10** using web browser application 280 to operate in accordance with the RUDP protocol to receive, by **client machine 10** from **remote machine 30’**, a responsive SYN segment (“packet”) during the setup of the RUDP connection. (EX1002 ¶¶181–182; *see also supra* Section IX.A.1.f.)

For example, “RUDP is designed to allow characteristics of each connection to be individually configured.” (EX1007, 17:42–46.) Thus, when a client initiates a connection, it first sends “a SYN segment which contains the negotiable parameters defined by the [Upper Layer Protocol] ULP via the [Application Programmer Interface] API.” (*Id.*, 24:40–42; *see also id.*, 18:56–62.) The server, upon receiving the client’s SYN segment, determines to either “accept” the proposed parameters in the received SYN segment or propose different parameters. (*Id.*, 24:42–44.) In either case, the server returns the parameters in its responsive SYN segment (“packet”) to the client. (*Id.*) The client, upon receiving the server’s SYN segment (“receiving a packet”), can then choose to accept the parameters sent by the server by responding to the server with an ACK message to establish the connection,

or it can send a reset (RST) segment to the server to refuse the connection. (*Id.*, 24:44–47; EX1002 ¶¶183–184.)

Berg explains that the messages sent to set up the RUDP connection, including the SYN segments, are UDP packets that include an RUDP header. (EX1007, 17:59–18:20, 19:1–40 (SYN segment); EX1002 ¶185.) Therefore, the SYN segment response sent from **remote machine 30'** and received by **client machine 10** in the *Wookey-Berg* combination is a packet (e.g., a UDP packet with an RUDP header). (EX1002 ¶¶185–187.)

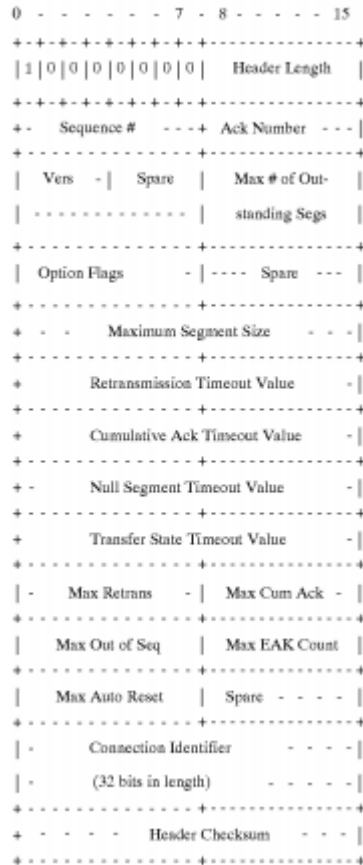
h) detecting an idle time period parameter field in the packet,

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶188–197.) For example, **client machine 10** in the combined *Wookey-Berg* system/method detects a null segment timeout parameter field (“an idle time period parameter field”) in the received SYN segment. (*Supra* Sections IX.A.1.f–g; EX1002 ¶189.)

For example, when the client receives the SYN segment sent from the server, it can *accept* or *reject* the proposed parameters (such as a null segment timeout value) in the received SYN segment. (EX1007, 24:44–48.) Figure 2, below, of the

Petition for *Inter Partes* Review
Patent No. 10,069,945

RUDP specification,¹⁶ illustrates the SYN segment with the respective fields for its negotiable parameters for the RUDP connection:



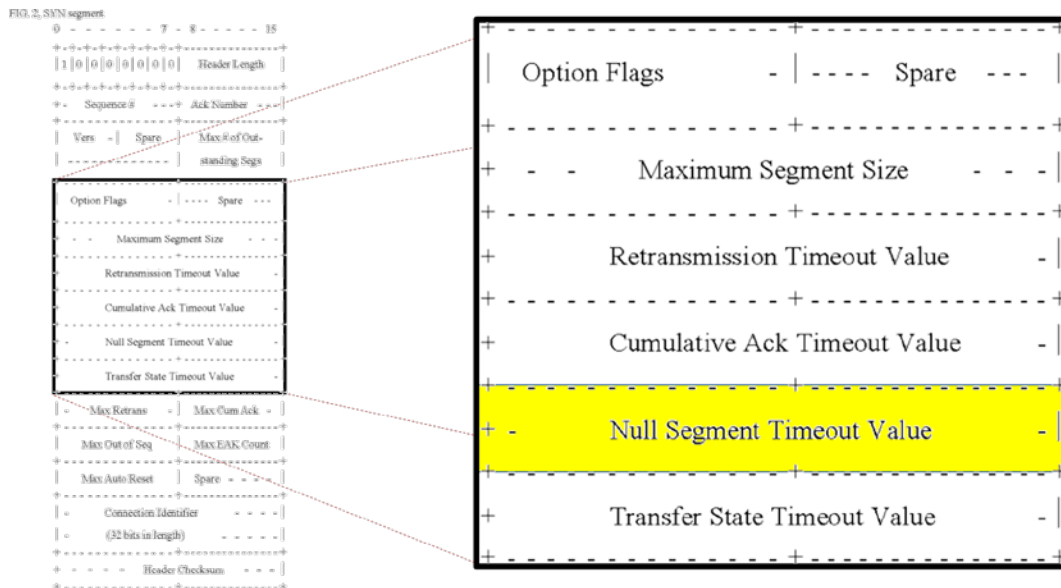
(*Id.*, 19:1–40; EX1002 ¶¶190–192.)

Berg further provides that the null segment timeout value is specified in milliseconds from a null segment timeout value (“idle time period”) parameter field

¹⁶ The RUDP specification is included in *Berg* as “Appendix 1.” (EX1007, 16:59–25:54; *see also* EX1011, 6.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

in the SYN segment. (EX1007, 18:58–60, 19:22–25, 20:36–42, 22:25–33.)



(*Id.*, 19:1–40 (excerpted and annotated); EX1002 ¶193.)

The *Wookey-Berg* combination would have thus been configured such that **client machine 10** detects the null segment timeout parameter field, in order to accept/reject the proposed null segment timeout value. (EX1002 ¶¶194–197; *see also supra* Section IX.A.1.f.ii, *infra* Section IX.A.1.i (discussing identification of the null segment timeout value in the SYN segment).)

Petition for *Inter Partes* Review
Patent No. 10,069,945

i) identifying metadata in the idle time period parameter field for an idle time period, where, after the idle time period is detected, the second protocol connection is deemed inactive,

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶198–212.) As discussed below, the *Wookey-Berg* combination discloses identifying a null segment timeout value (“metadata”) in the null segment timeout parameter field for a null segment timeout (“idle time period”), where, after the null segment timeout is detected, the RUDP connection is deemed inactive. (*Id.* ¶¶199–200.)

For example, the client detects the null segment timeout parameter field in the SYN segment sent from the server so it can *accept* or *reject* the proposed null segment timeout value contained therein. (*See supra* Section IX.A.1.h.) Thus, in order to *accept* or *reject* the proposed null segment timeout value in the null segment timeout parameter field, the client necessarily identifies it. (EX1002 ¶¶201–202.)

To the extent the *Wookey-Berg* combination does not disclose the claimed “identifying,” it would have been rendered obvious in the combined *Wookey-Berg* system/process. (*Id.* ¶203.) A POSITA would have been motivated to configure the *Wookey-Berg* system/process to include functionality that identifies this parameter to facilitate the subsequent process of accepting or rejecting it in a manner consistent with the features described by *Berg*. (EX1007, 24:44–48; EX1002 ¶¶204–206.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

The '945 patent confirms the understanding that the null segment timeout value is metadata, explaining that metadata can be a value indicating a “duration of time” (EX1001, 21:60–65), which is what the above-discussed timeout value in *Berg* contains. (EX1007, 20:36–47; EX1002 ¶207.) Moreover, the above understanding is consistent with PO’s view of the same claimed features. (EX1017, 13 (PO alleging “identifying metadata (e.g., a value, etc.) in the idle time period parameter field (e.g., idle_timeout parameter field, etc.)”).)

The *Wookey-Berg* combination further discloses that the null segment timeout value is for an “idle time period, where, after the idle time period is detected, the second protocol connection is deemed inactive.” For instance, the null segment timeout value indicates the amount of time (e.g., milliseconds) to wait before sending a null segment when the connection has been idle (e.g., when a data segment has not been sent or received). (EX1007, 20:36–40.)¹⁷ The null segment is sent to determine “if the other side of a connection is still active.” (*Id.*, 22:25–33, 20:3–40.) This waiting period (when the connection is idle) before the client sends a null

¹⁷ *Berg* uses “null segment,” “NUL segment,” and “keep-alive” interchangeably. (EX1007, 20:36–42, 22:25–33; EX1002 ¶209.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

segment to the server to verify whether the connection is active represents an “idle time period.” (EX1002 ¶208.)

Furthermore, as a result of detecting a null segment timer’s expiration, which is set based on the null segment timeout value, the RUDP connection is “deemed inactive,” as claimed. (EX1002 ¶209.) For example, *Berg* explains that the null segment mechanism’s purpose is to determine whether the “connection is *still active*.” (EX1007, 22:25–32.) *Berg*’s null segment mechanism utilizes a null segment timer to control when to send a null segment. (*Id.*, 23:45–49.) For example, the client will send a null segment to the server when the client’s null segment timer expires. The duration of the client’s null segment timer is set to the null segment timeout (“idle time period”) specified by the null segment timeout value. (*Id.*, 23:45–49.) That is, the null segment timeout value defines a duration of time that the connection will remain inactive before the client sends a null segment to see if the connection still exists. (*Id.*, 20:36–41, 22:25–32; EX1002 ¶210.) Accordingly, the identified null segment timeout value is for a null segment timeout, where, in response to detecting expiration of the null segment timer (which was set to the null segment timeout), the RUDP connection is deemed inactive (and thereby causing the client to send a null segment to the server to see if the connection is still active). (EX1002 ¶210.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been further configured to include **client machine 10** identifying the null segment timeout value in the null segment timeout parameter field for a null segment timeout, where, in response to expiration of the null segment timeout being detected, the RUDP connection is deemed inactive, as set forth in limitation 1.i. (EX1002 ¶¶211–212; *see supra* Section IX.A.1.f.ii.)

j) creating or modifying, by the client computer and based on the metadata, a timeout attribute associated with the second protocol connection.

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶213–217.) As explained above and further explained below, **client machine 10** in the *Wookey-Berg* combination would have been configured to create or modify, based on the null segment timeout value (“metadata”), a null segment timer (“timeout attribute”) associated with the RUDP connection. (EX1002 ¶214; *see supra* Section IX.A.1.i.)

For example, the client determines the timeout value for a null segment timer associated with the RUDP connection based on the null segment timeout value. (EX1007, 20:36–38, 23:46–59; EX1002 ¶215; *see supra* Section IX.A.1.i.) As discussed above, the client’s null segment timer is set (“creat[ed] or modif[ied]”)

based on the null segment timeout value. (EX1007, 23:45–59; *see also id.*, 20:36–41; EX1002 ¶215; *supra* Section IX.A.1.i.)

Accordingly, in the *Wookey-Berg* combination, **client machine 10** creates or modifies, based on the null segment timeout value (“metadata”), a null segment timer (“timeout attribute”) associated with the RUDP (“second protocol”) connection, as set forth in limitation 1.j. (EX1002 ¶¶216–217; *see supra* Section IX.A.1.f.ii.)

2. Claim 9

a) The computer-implemented method of claim 1 wherein the code causes the client computer to operate such that the packet is received before the second protocol connection is established.

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for at least reasons given above for limitations 1.f–g. (EX1002 ¶¶218–221.) As explained above, the code (e.g., the encoded URLs in HTML page 288 or HTML page 288 including the encoded URLs) causes **client machine 10** to initiate the process of establishing an RUDP connection between **client machine 10** and **remote machine 30’**. (*See supra* Sections IX.A.1.f–g.) The process of establishing the RUDP connection involves **client machine 10** receiving the SYN segment including connection parameters such as the null segment timeout value, and upon completion of the negotiation (e.g., when the connection parameters

Petition for *Inter Partes* Review
Patent No. 10,069,945

are acceptable to both nodes), the RUDP connection is established. (*Id.*; EX1007, 24:39–47 (describing that based on the parameters included in the server’s responsive SYN segment, which may be the same or different than those initially proposed by the client, “[t]he client can then choose to accept the parameters send [sic] by the server *by sending an ACK to establish the connection* or it can send a RST to refuse the connection.”).) Thus, in the *Wookey-Berg* combination, the code causes **client machine 10** to operate such that the SYN segment is received before the RUDP connection is established. (EX1002 ¶¶219–221.)

3. Claim 10

a) The computer-implemented method of claim 1 wherein the code causes the client computer to operate such that, after the second protocol connection is established, another packet is received with the idle time period parameter field.

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶222–229.) For example, as discussed above, the SYN segment received by **client machine 10** during the RUDP connection establishment process includes a null segment timeout parameter field (“idle time period parameter field”). (*See supra* Sections IX.A.1.f–h, IX.A.2.)

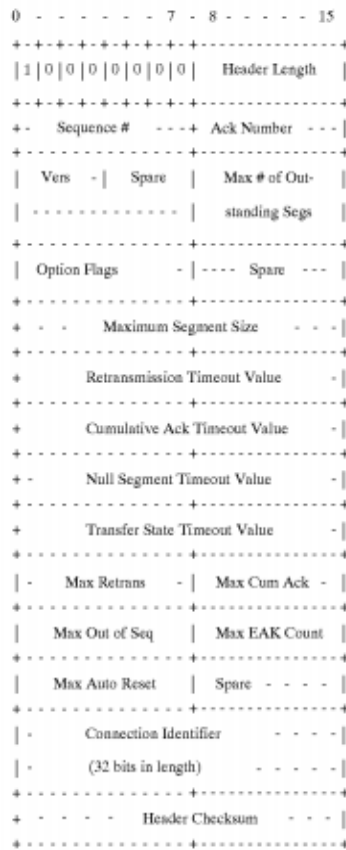
The *Wookey-Berg* combination further discloses and/or suggests that this SYN segment including the null segment timeout parameter field is received by **client machine 10** after the RUDP connection is established, as claimed. In

Petition for *Inter Partes* Review
Patent No. 10,069,945

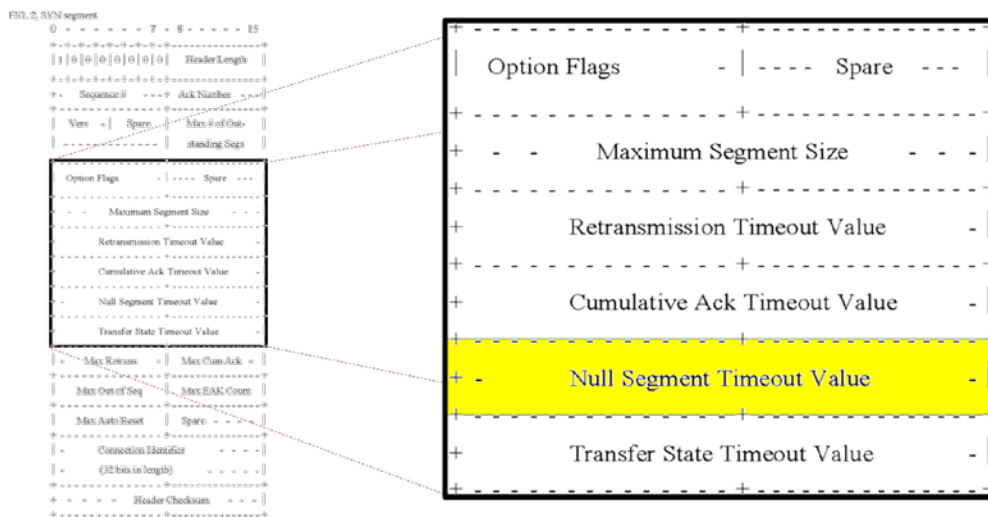
particular, *Berg* discloses that in addition to “[t]he SYN [being] used to establish a connection” as discussed above, the “*SYN segment is also used to perform an auto reset on a connection.*” (EX1007, 18:56–67.) An auto reset is performed to reset an already-established RUDP connection because if the connection was not already established, there would be no connection to “reset.” (EX1002 ¶224; EX1007, 21:24–34 (describing that a unique connection ID is saved by both nodes of the RUDP connection and “[w]hen an auto reset is performed, the peer shall send *the saved connection ID originally received* to indicate that an auto reset is being performed *on the connection.*”), 23:45–59 (describing the null segment mechanism implemented in an established connection, where an auto reset on the connection is initiated after the null segment timer expires a certain number of times).)

As discussed above for claim limitation 1.h, the SYN segment includes a null segment timeout parameter field (“idle time period parameter field”). (*See supra* Section IX.A.1.h.)

Petition for *Inter Partes* Review
Patent No. 10,069,945



(EX1007, 19:1–40 (SYN segment).)



(*Id.*, 19:1–40 (excerpted and annotated); EX1002 ¶225.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

Berg discloses that in the case of an auto reset, an options flag field named “REUSE” in the SYN segment can be set to indicate that the previous negotiable parameters should be used. (EX1007, 19:63–20:15.) When the “REUSE” options flag is not set, then all the fields of a SYN segment (including the null segment timeout parameter field) are present in the SYN segment. (*Id.*; EX1002 ¶226.) If the “REUSE” options flag is set, *Berg* discloses that certain specified fields are absent from the SYN segment, namely “Maximum Segment Size, Retransmission Timeout Value, Cumulative Ack Timeout Value, Max Retrans, Max Cum Ack, Max Out of Seq, and Max EAK Count.” (EX1007, 20:9–14.) Thus, even in the latter case, the SYN segment includes the null segment timeout parameter field, as it is not one of the fields specified as being absent from the SYN segment. (*Id.*; EX1002 ¶226.)

Berg further discloses that the client receives the SYN segment from a server initiating an auto reset. *Berg* explains that “[e]ither side of a connection can initiate an auto reset,” and thus if the server initiates the auto reset, it would send a SYN segment to the client such that the client would receive the SYN segment with the null segment timeout parameter field. (EX1007, 23:60–24:4, 22:44–64; EX1002 ¶227.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been configured such that the code causes **client machine 10** to operate such that, after the RUDP connection is established, another SYN segment is received with the null segment timeout parameter field, e.g., to perform an auto reset on the established RUDP connection. (EX1002 ¶¶228–229.)

4. Claim 34

a) A computer-implemented method, comprising:

To the extent limiting, *Wookey* discloses the limitations of this preamble for at least the same reasons as above for limitation 1.a. (*See supra* Section IX.A.1.a; EX1002 ¶¶230–231; *infra* Sections IX.A.4.b–i.)

b) providing access to a server computer including:

- (1) a non-transitory memory storing a network application, and
- (2) one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application to operate in accordance with a first protocol including a transmission control protocol (TCP);

This limitation 34.b varies slightly from limitation 1.b in that it recites “providing access to a server computer” while limitation 1.b recites “causing access to be provided to a server computer,” where the claimed server computer includes the same components. (*Supra* Section IX.A.1.b.) *Wookey* discloses limitation 34.b

Petition for *Inter Partes* Review
Patent No. 10,069,945

for at least the same reasons discussed above for limitation 1.b because by causing access to be provided to **remote machine 30** (“server computer”) as discussed above for limitation 1.b, *Wookey* discloses providing access to **remote machine 30**. (*Supra* Section IX.A.1.b; EX1002 ¶232.)

- c) causing a TCP connection to be established with a client computer, by:
 - communicating a segment including at least one first synchronize bit;
 - communicating a first acknowledgement of the segment, and at least one second synchronize bit; and
 - communicating a second acknowledgement;**
- d) causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer;¹⁸**
- e) causing the server computer to permit second data, from the user of the client computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP);¹⁹ and**

Wookey discloses limitations 34.c–e for at least the same reasons discussed above for limitations 1.c–e. (*Supra* Sections IX.A.1.c–e; EX1002 ¶¶233–235.)

¹⁸ *See supra* n.9.

¹⁹ *See supra* n.10.

Petition for *Inter Partes* Review
Patent No. 10,069,945

f) providing access to code that, after use by the client computer, results in the client computer operating in accordance with a second protocol that is separate from the TCP, in order to establish a second protocol connection with another server computer, by:²⁰

The *Wookey-Berg* combination discloses and/or suggests limitation for at least the same reasons discussed above for limitation 1.f. (*Supra* Section IX.A.1.f.) For example, as discussed above, the *Wookey-Berg* combination method provides access to code that, after use by **client machine 10** (e.g., by clicking on one of the encoded URLs in HTML page 288), results in **client machine 10** operating in accordance with a RUDP protocol (“second protocol that is separate from the TCP”), in order to establish an RUDP connection (“second protocol connection”) with **remote machine 30’** (“another server computer”). (*Id.*; EX1002 ¶236.)

g) identifying idle information for detecting an idle time period, after which, the second protocol connection is subject to deactivation,

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶237–250.) As discussed above, the *Wookey-Berg* combination would have included processes that involved establishing an RUDP connection between **client machine 10** and **remote machine 30’**. (*Supra* Sections

²⁰ See *supra* n.11.

Petition for *Inter Partes* Review
Patent No. 10,069,945

IX.A.4.f; IX.A.1.f.) As discussed above for limitation 1.i and further below, *Berg*'s RUDP protocol includes processes for identifying a null segment timeout value ("idle information") for detecting a null segment timeout ("idle time period"), after which, the RUDP connection ("second protocol connection") is subject to deactivation. (*Supra* Section IX.A.1.i; *see also supra* Sections IX.A.4.f; IX.A.1.f; EX1002 ¶238.)

For example, when setting up a RUDP connection, the nodes will negotiate a null segment timeout value. (*Supra* Sections IX.A.1.f–j.) The null segment timeout value, which is "idle information," is for detecting an idle time period that results in the RUDP connection being subject to deactivation. (*Supra* Section IX.A.1.i; EX1002 ¶239.)

This understanding of "idle information" is consistent with the '945 patent, which provides that "[i]dle information ... may include ... a duration of time ... [which] may be specified according to various measures of time including seconds." (EX1001, 12:40–44; EX1002 ¶240.)

Furthermore, as a result of detecting a null segment timer's expiration, which is set based on the negotiated null segment timeout value, the RUDP connection is "subject to deactivation," as claimed. (EX1002 ¶241.) The null segment timeout value is used to set the duration of a client's null segment timer "for sending

Petition for *Inter Partes* Review
Patent No. 10,069,945

a null segment *if a data segment has not been sent.*” (EX1007, 20:36–42, 23:45–49, 22:25–32.) Upon detecting the null segment timer’s expiration, the client sends a null segment to the server to determine if the connection is still valid. (*Id.*, 23:45–59.) When the server receives a null segment, it “*must* acknowledge the segment if a valid connection exists.” (*Id.*, 22:27–30.) If the null segment is not acknowledged, the connection is subject to being closed via an “auto reset.” (*Id.*, 23:45–59; EX1002 ¶241; *see also supra* Section IX.A.3.)

The “auto reset,” which is triggered upon the null segment timer’s expiration, is a form of deactivation of the RUDP connection. (EX1002 ¶242; EX1007, 21:15–23, 23:60–24:4.) For example, an “auto reset” signals a problem with the connection and that the connection needs to be deactivated. (EX1007, 23:60–24:4; *see also id.*, 21:15–23, 24:15–25.) The auto-reset involves sending a reset (RST) segment to close or reset the connection. (*Id.*, 23:60–24:4; *see also id.*, 18:27–28, 22:20–24.) Either closing or resetting a connection is a form of deactivation for the connection. (EX1002 ¶¶242–243.)

Regarding the “*identifying* idle information,” when the client initiates a connection, the client’s ULP, e.g., an application, identifies negotiable parameters that indicate the client’s desired features for the RUDP connection being established. (EX1007, 24:40–48; *see also id.*, 18:57–64.) Because the client must select and

Petition for *Inter Partes* Review
Patent No. 10,069,945

include the desired negotiable parameters (including the null segment timeout value) in the SYN segment to be sent to the server, the client identifies the null segment timeout value. (EX1002 ¶244.)

Also, the client identifies these parameters when the server initiates the connection. For example, when the server initiates a reset (EX1007, 23:61–65), the server sends an initial SYN segment (*id.*, 18:66–67) to the client to negotiate the connection. The client can then choose to “accept these parameters ... or propose different parameters” in its responsive SYN message. (*Id.*, 24:43–45.) Thus, the client necessarily identifies the null segment timeout value in the SYN segment, one of the parameters, upon receiving the SYN segment. (EX1002 ¶¶245–247; EX1007, 18:58–61; *see also supra* Section IX.A.1.i.)

To the extent the *Wookey-Berg* combination does not disclose the claimed “identifying,” it would have been rendered obvious in the *Wookey-Berg* system/process. (EX1002 ¶248.) A POSITA would have been motivated to configure the *Wookey-Berg* system/process to include functionality for identifying the parameters, including null segment timeout value, to facilitate the subsequent process of (i) including the client’s desired parameters for the RUDP connection in the SYN segment to be sent or (ii) accepting/rejecting the parameter in the received SYN packet. (EX1007, 24:44–48.) Therefore, the above-discussed *Wookey-Berg*

Petition for *Inter Partes* Review
Patent No. 10,069,945

system/process discloses and/or suggests such features to the extent not disclosed. (EX1002 ¶248.)

Accordingly, for reasons similar to those explained above, a POSITA would have been motivated to implement in the *Wookey-Berg* combined process, the process of identifying, by the ULP/application of **client machine 10** and/or from the received SYN segment from **remote machine 30'**, a null segment timeout value for detecting a null segment timeout, after which, the RUDP connection is subject to deactivation. (*See supra* Section IX.A.1.f.ii; EX1002 ¶249.)

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 34.g. (EX1002 ¶250.)

h) generating a second protocol packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information, and

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶251–259.) As discussed below, the RUDP protocol in the *Wookey-Berg* combination would have involved generating a SYN segment (“second protocol packet”) including a null segment timeout value field (“idle time period parameter field”) identifying a null segment timeout value (“metadata”) for the null segment timeout (“idle time period”) based on the identified null segment

Petition for *Inter Partes* Review
Patent No. 10,069,945

timeout value (“idle information”). (EX1002 ¶¶252–253; *supra* Sections IX.A.1.f–i.)

For example, *Berg* discloses generating a SYN segment (“second protocol packet”) for establishing an RUDP connection. (EX1002 ¶254.) As explained above, the RUDP connection is established by the exchange of RUDP SYN segments, which includes parameter fields for all of the proposed negotiable parameters for the RUDP connection, between the two hosts. (EX1007, 18:57–61, 24:40–47; 24:40–48; *supra* Sections IX.A.1.g–h.) Further, as explained for limitations 1.h–i and limitation 34.g, the SYN segment includes a null segment timeout value field (“idle time period parameter field”) for identifying the null segment timeout value (“metadata”) that is specified in milliseconds based on the identified null segment timeout value (“idle information”). (*Supra* Sections IX.A.1.h–i, IX.A.4.g; EX1002 ¶¶254–255.)

Berg discloses this generating limitation in at least two ways. (EX1002 ¶256.) First, when the client generates the initial SYN segment, the ULP/application of the client identifies the negotiable null segment timeout value (“idle information”), for the SYN segment via an API. (EX1007, 24:40–48; *see also id.*, 18:57–61; *supra* Section IX.A.4.g.) The client then “initiates a connection” by sending the generated

Petition for *Inter Partes* Review
Patent No. 10,069,945

“SYN segment which contains the negotiable parameters” to the server. (EX1007, 24:40–47; EX1002 ¶256.)

Second, when the client receives the initial SYN segment from the server, the client chooses to “accept these parameters by echoing them back in its SYN message or propose different parameters in its SYN response,” i.e., the client generates a responsive SYN segment based on the identified null segment timeout value in the initial SYN segment. (EX1007, 24:43–45; EX1002 ¶257; *supra* Sections IX.A.4.g, IX.A.1.i.)

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been further configured to include generating, by **client machine 10**, a SYN segment including a null segment timeout value field identifying the null segment timeout value for the null segment timeout based on the identified null segment timeout value. (EX1002 ¶258.) Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 34.h. (EX1002 ¶259.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

i) **sending, from the client computer to the another server computer, the second protocol packet to provide the metadata for the idle time period to the another server computer, for use by the another server computer in creating or modifying, based on the metadata, a timeout attribute associated with the second protocol connection.**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶260–265.) As discussed below, the RUDP protocol in the *Wookey-Berg* combination would have involved sending, from **client machine 10** to **remote machine 30'**, the SYN segment to provide the null segment timeout value, for the null segment timeout to **remote machine 30'**, for use by **remote machine 30'** in creating or modifying, based on the received null segment timeout value, a null segment timer (“timeout attribute”) associated with the RUDP connection. (EX1002 ¶261; *supra* Sections IX.A.1.f–i.)

For example, *Berg* disclose sending, from the client to the server, the SYN segment to provide the null segment timeout value, for the null segment timeout to the server in at least two ways. (EX1002 ¶262; *supra* Section IX.A.4.h.) First, when the client initiates an RUDP connection, the client sends the initial SYN segment “which contains the negotiable parameters” including the identified null segment timeout value to the server. (EX1007, 24:40–47.) Second, when the client receives the initial SYN segment from the server, the client can then choose to “accept these

Petition for *Inter Partes* Review
Patent No. 10,069,945

parameters by echoing them back in its SYN message or propose different parameters in its SYN response.” (*Id.*, 24:43–45; EX1002 ¶262.)

In either scenario, the server creates or modifies its own null segment timer (“timeout attribute”), based on the negotiated null segment timeout value. For example, the server’s null segment timer’s duration is created or modified by setting the duration to twice the client’s null segment timeout value, when the null segment timeout value is not zero. (EX1007, 23:45–52.) On the other hand, when the null segment timeout value is zero, the server’s null segment timer is disabled. (EX1007, 23:45–59.) Thus, the server creates or modifies, based on the client’s null segment timeout value, the server’s null segment timer associated with the RUDP connection. (EX1002 ¶263.)

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have includes the process of sending, from **client machine 10** to **remote machine 30’**, the RUDP SYN segment to provide the null segment timeout value for the null segment timeout to **remote machine 30’**, for use by **remote machine 30’** in creating or modifying, based on the null segment timeout value, a null segment timer associated with the RUDP connection. (EX1002 ¶¶264–265.)

5. Claim 64**a) The computer-implemented method of claim 34 wherein the access to the code is provided via hypertext.²¹**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶266–275.) As discussed below, the HTML page 288 and the contents therein (including the encoded URLs) are hypertext that **client machine 10**’s HTTP-based web browser application 280 uses to display the HTML page 288 so the user of **client machine 10** can select and access available resources on the server farm 38, thereby providing access to the code via hypertext, as claimed. (*Id.*)

For example, **remote machine 30** in the *Wookey-Berg* combination provides access to **client machine 10** to the *HyperText* Markup Language (HTML) page 288 over a *HyperText* Transfer Protocol (HTTP) to **client machine 10**. (*See supra* Sections IX.1.A.1.d–f, IX.A.1.4.d–f.) The web browser application 280 on **client machine 10** displays the received HTML page 288 including selectable icons 257/257’ (an example of “hypertext”) in the HTML page 288 such that **client machine 10** can access the code (e.g., the encoded URLs in HTML page 288 or HTML page 288 including the encoded URLs). (EX1002 ¶¶268–270; EX1005

²¹ *See supra* n.11.

Petition for *Inter Partes* Review
Patent No. 10,069,945

¶¶[0196], [0216]; *see also id.* ¶¶[0757]–[0758], [0764]–[0765] (explaining that the graphical or textual link displayed on the web page is a hyperlink).)

The displayed HTML page 288 is hypertext because it displays information with selectable resources via the graphical icons 257/257’ (e.g., hyperlinks), associated with the encoded URLs, that the user may access. (EX1002 ¶¶271–272 (citing EX1032–EX1033).) The displayed HTML page 288 is also hypertext because it is transmitted over a HTTP protocol and displayed in an HTTP web browser 280. (EX1002 ¶273.)

This understanding regarding hypertext is also consistent with PO’s view of the same claimed features. (EX1017, 47 (PO alleging “hypertext (e.g., a hypertext transfer protocol (HTTP) link, etc.)”).)

Accordingly, the *Wookey-Berg* combination discloses and/or suggests that the access to the code is provided via the displayed HTML page 288 (“hypertext”) on **client machine 10**’s web browser application 280. (EX1002 ¶274.)

Thus, the *Wookey-Berg* combination discloses and/or suggests claim 64. (EX1002 ¶275.)

6. Claim 65

a) The computer-implemented method of claim 64 wherein the hypertext includes a hypertext transfer protocol (HTTP) link.

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶276–279.) For example, as discussed above, the displayed HTML page 288 (“hypertext”) includes selectable icons 257, 257’ that are each associated with an encoded URL that specifies a resource. (*Supra* Sections IX.A.4.f, IX.A.5.) The displayed HTML page includes HTTP links at least in the form of the displayed selectable icons associated with the encoded URLs. (*Id.*; EX1002 ¶277.) This is because **client machine 10** receives the HTML page 288 over an *HTTP* protocol, and each encoded URL specifies “*the location of the resource ...; [and] a launch command associated with the resource,*” which is what an HTTP link would have specified in an HTTP-enabled browser. (EX1005 ¶[0216]; *see also id.* ¶¶[0757]–[0758], [0764]–[0765] (explaining that the graphical or textual link displayed on the web page is a hyperlink); EX1002 ¶¶277–278 (citing EX1032–EX1033).)

Thus, the *Wookey-Berg* combination discloses and/or suggests claim 65. (EX1002 ¶279.)

7. Claim 66

a) The computer-implemented method of claim 34 wherein the code is configured to be used by the client computer via hypertext.²²

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶280–282.) As discussed above for claim 64, the HTML page 288 including the encoded URLs is configured to be displayed for selection of a desired resource (i.e., the “code is configured to be used”) on **client machine 10**’s web browser application 280 via the contents (“hypertext”) included in HTML page 288, which specify how the HTML page 288 should be displayed. (*Supra* Sections IX.A.1.f, IX.A.4.f, IX.A.5; EX1005 ¶¶[0196], [0216]; EX1002 ¶¶281–282.)

8. Claim 67

a) The computer-implemented method of claim 66 wherein the hypertext includes a hypertext transfer protocol (HTTP) link.

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for at least the same reasons discussed above for claim 65. (*Supra* Section IX.A.6; EX1002 ¶283.)

²² See *supra* n.11.

9. Claim 68

a) The computer-implemented method of claim 34 wherein the access to the code is provided via the hypertext transfer protocol (HTTP).²³

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶284–286.) For example, as discussed above for claim 64, the **client machine 10**’s ability to access the code (e.g., the encoded URLs in HTML page 288 or HTML page 288 including the encoded URLs) is provided via hypertext transfer protocol (HTTP) because the **client machine 10** would display the received HTML page 288 in **client machine 10**’s HTTP-enabled browser 280, and also because the **client machine 10** received the HTML page 288 over an HTTP protocol. (*See supra* Sections IX.1.A.4.f, IX.1.A.5.)

10. Claim 69

a) The computer-implemented method of claim 68 wherein the hypertext transfer protocol (HTTP) transports hypertext that includes a hypertext transfer protocol (HTTP) link.

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶287–288.) For example, as discussed above, the HTTP protocol is used to transmit/receive (“transport[]”), e.g., over communication link

²³ *See supra* n.11.

Petition for *Inter Partes* Review
Patent No. 10,069,945

150, the contents of HTML page 288 (“hypertext”) that includes the selectable icons associated with the encoded URLs (“hypertext transfer protocol (HTTP) link”). (*See supra* Sections IX.1.A.1.f, IX.1.A.4.f, IX.1.A.5–9.)

X. DISCRETIONARY DENIAL IS NOT APPROPRIATE

The Board should not exercise its discretion to deny institution under 35 U.S.C. § 314(a). *NHK Spring Co., Ltd. v. Intrix-Plex Techs, Inc.*, IPR2018-00752, Paper 8 (Sept. 12, 2018) does not apply here as no trial date has been scheduled in the related litigation. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 3 (Mar. 20, 2020) (precedential) (“*NHK* applies ... where the district court has set a trial date to occur earlier than the Board’s deadline to issue a final written decision in an instituted proceeding.”). The six-factor test addressed in *Fintiv* (“*Fintiv* factor(s)”) favors institution. *See id.*, 5–6.

The **first *Fintiv* factor** (stay) is neutral, if not weighing in favor of institution. While there is currently no stay pending IPR resolution before the Western District of Texas (“WDTX”) court, Google intends to pursue such a stay should institution be granted. The Board should not speculate as to how the court would rule on the requested stay based on actions taken in different cases. *Western Digital Corp. et al. v. Martin Kuster*, IPR2020-01391, Paper 10 at 8–9 (February 16, 2020) (finding this factor neutral despite Patent Owner assertions regarding WDTX Judge

Petition for *Inter Partes* Review
Patent No. 10,069,945

Albright’s past practices regarding stays). Moreover, Google filed a motion to transfer in the district court litigation. The Court has currently stayed the litigation pending resolution of Google’s transfer motion. (EX1030.)

The **second *Fintiv* factor** (proximity of trial dates) weighs strongly in favor of institution. *First*, the district court has not set a trial date, which “weighs significantly against exercising [] discretion to deny institution of the proceeding.” *Google LLC v. Uniloc 2017 LLC*, IPR2020-00441, Paper 13 at 35 (July 17, 2020); *Nested Bean, Inc. v. Big Beings USA PTY LTD*, IPR2020-01234, Paper 15 at 14 (Jan. 25, 2021) (same). In fact, while the parties submitted proposed schedules, the court granted Google’s motion to stay the case pending resolution of Google’s transfer motion and declined to set a case schedule.²⁴ (EX1030; EX1024.) Accordingly, determining at this time whether the Board’s anticipated FWD date (e.g., October–November 2022) will occur after trial in the related litigation would be speculative.

Even if the parties’ most recent proposed schedule was considered as a baseline for comparison (which Google submits would be improper at this time),

²⁴ Disposition of Google’s transfer motion has taken priority over other activities. (EX1030.) *See also In re Apple Inc.*, 979 F.3d 1332, 1337 (Fed. Cir. 2020) (“disposing of a filed transfer motion ‘should unquestionably take top priority.’”).

Petition for *Inter Partes* Review
Patent No. 10,069,945

dates in that proposed schedule would need to be delayed because they were proposed prior to the court's recent stay order and did not take the stay into consideration. (EX1030; EX1024, 5–9.) Accordingly, the proposed dates in light of the court's stay order demonstrate that trial will likely occur after August 2022.²⁵ (EX1024, 6, 9.) Thus, any projected trial date (*albeit* speculative) would be more likely after the expected due date of the Board's FWD (e.g., around October–November 2022).

The Board has instituted trial even where an *actual* litigation trial date precedes the FWD deadline by two months or more. *Sand Revolution II, LLC v. Cont'l Intermodal Grp–Trucking LLC*, IPR2019-01393, Paper 24 at 8–10 (June 16, 2020) (“*Sand*”) (informative) (factor 2 favored institution despite WDTX trial preceding FWD deadline by five months); *Western Digital Corp.*, IPR2020-01391

²⁵ Consistent with the court's practice, the proposed order stages deadlines and trial date based on the proposed August 2021 *Markman* hearing and that the “actual trial date” may materially differ from the schedule and that the court would consider “reasonable amendments to the case schedule post-*Markman*.” (EX1024, 9 n.3; *id.*, 6.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

at 9-10; *Apple Inc. v. Parus Holdings, Inc.*, IPR2020-00686, Paper 9 at 11–13, 22; *Fintiv*, IPR2020-00019, Paper 11 at 5, 9.

It is also unlikely that trial in WDTX will proceed without delay. WDTX has issued a suspension order every month for the past twelve months suspending almost all trials in the district from March 13, 2020 to at least April 30, 2021 due to the COVID-19 pandemic, creating a large backlog of trials. (EX1025; *see also* EX1026, 2 (agreeing to stipulation to postpone trial date “to allow the COVID-19 situation to ameliorate”); EX1029 (showing J. Albright’s docket including 792 cases filed in 2020).) Thus, as the Board has recognized, civil trials in WDTX “may possibly slip ... [due to] months of backlogged trials, including many active criminal cases that would take precedence over civil trials.” *HP Inc. v. Slingshot Printing LLC*, IPR2020-01085, Paper 12 at 7 (Jan. 14, 2021); *Sand* at 8-10.

Finally, the potential for transfer based on Google’s transfer motion further reduces the likelihood that trial will precede the Board’s FWD. *Dish Network, L.L.C. v. Broadband iTV, Inc.*, IPR2020-01267, Paper 15 at 17–18 (Jan. 21, 2021) (explaining that the second *Fintiv* factor “is a proxy for the *likelihood* that the trial court will reach a decision on validity issues before the Board reaches a [FWD]”). Indeed, transfer of the WDTX case to the proper venue will undoubtedly delay any trial beyond the anticipated FWD due date. *C.f. Google LLC v. Uniloc 2017 LLC*,

Petition for *Inter Partes* Review
Patent No. 10,069,945

IPR2020-00115, Paper 8, at 7 n.7 (Mar. 27, 2020) (denying institution based in part on scheduled trial date that was later vacated upon transfer); *see Uniloc 2017 LLC v. Google LLC*, Case No. 18-CV-00502, Dkt No. 277 (E.D. Tex. June 19, 2020) (transfer order vacating unreach deadlines).

The **third *Fintiv* factor** (investment in parallel proceedings) also weighs in favor of institution. The case is in its earliest stages, and as discussed, all deadlines other than those concerning Google’s transfer motion have been stayed. Thus, investment by the parties in invalidity and claim constructions issues will be delayed, especially if the case is transferred. And even if the transfer motion is denied, using the above-discussed proposed schedule as an exemplary baseline (which will require adjustments), any *Markman* hearing in the case is likely not to occur until August 2021 or later, and thus would occur near the expected due date of the Board’s institution decision. (EX1024, 6.) Discovery will not commence, and dispositive motions will not be due, until after the *Markman* hearing. Other case activity that has taken place or will eventually take place (outside claim construction) that does not relate to validity should “not weigh in [the Board’s] consideration of this issue.” *Western Digital Corp.*, IPR2020-01391 at 11. Such minimal investment is insufficient to support discretionary denial. *Juniper Networks, Inc. v. Huawei Digital Techs. (Cheng Du) Co., Ltd.*, IPR2020-01130, Paper 13 at 12–13 (Jan. 22,

Petition for *Inter Partes* Review
Patent No. 10,069,945

2021); *Dish* at 19–21; *HP* at 7. Additionally, Google’s diligence in filing this Petition just four months after receiving PO’s narrowed list of asserted claims²⁶ further weighs against discretionary denial. *Dish* at 20–21; *Fintiv* at 11.

The **fourth *Fintiv* factor** (overlap) is neutral if not in favor of institution. There is, at this early stage of the district court litigation, no evidence of overlap with the grounds here and any invalidity positions to be pursued in the district court. Independent claims 1 and 34 being challenged here are not asserted in the litigation. Moreover, preliminary invalidity contentions have not yet been served. And, as discussed above, the district court must first resolve the pending transfer motion before addressing substantive issues, including invalidity. Given the stay of the litigation, it is likely expert discovery relating to invalidity will not be completed

²⁶ PO initially asserted 455 claims across eight patents (including 141 claims from the ’945 patent). (EX1035, 1.) The WDTX court ordered PO to substantially reduce the asserted claims to approximately 65 claims. On October 20, 2020, PO narrowed its asserted claims as ordered, but “reserve[d] the right to either narrow the claims further, **substitute claims**, or **add a reasonable number of claims**.” (EX1027.) On December 4, 2020, PO informed Petitioner that it will not modify its narrowed list of asserted claims “**at this time**.” (EX1028.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

until at the earliest mid-2022 and dispositive motions on those issues would be due after that. Accordingly, substantive invalidity issues would not be addressed in the litigation until well past the institution decision and likely very near the time of the Board's expected FWD. Moreover, Petitioner may rely on invalidity grounds not asserted here in the district court litigation, including system art and prior art references and combinations not raised herein. Therefore, any potential overlap between this proceeding and the district court is minimal and speculative at this time.

Other circumstances (***Fintiv* factor six**) also favor institution. The Petition presents strong grounds demonstrating the unpatentability of the challenged claims, which weighs against discretionary denial. (*See supra* Section IX.) *Western Digital Corp.*, IPR2020-01391 at 14-15. Indeed, the Board has already instituted the Unified -742 IPR (based on *Berg*) challenging claims of the related '995 patent. (*See supra* Section II.) *Berg* is being applied here. Moreover, this petition and the concurrently-filed sister petition (*id.*) are the only challenges to the '945 patent before the Board, which is a "crucial fact" favoring institution. *Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 10 at 6 (May 12, 2020).²⁷

²⁷ Discretionary denial under *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper No. 19 is not applicable here given the

Petition for *Inter Partes* Review
Patent No. 10,069,945

While ***Fintiv* factor 5** (same parties) may weigh slightly in favor of denial, the remaining factors are at least neutral, if not strongly favoring institution. Further, even if the Board finds that **factor 1** (or even **factor 2**) somehow tilts in favor of denial, Petitioner’s diligence, the lack of relevant investment of resources, lack of evidence of overlap, and/or strength of Petitioner’s grounds (**factors 3, 4, 6**) outweigh these other factors. *See e.g., SK Hynix Inc. et al. v. Netlist, Inc.*, IPR2020-01421, Paper 10 at 6–13 (Mar. 16, 2021). Accordingly, based on a “holistic view” of whether integrity of the system and efficiency is best served, institution here is proper. *Samsung Elecs. Co. Ltd. v. Dynamics Inc.*, IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

’945 patent is not at issue in any other proceeding pending before the Board (outside of Petitioner’s concurrently-filed sister petition).

Petition for *Inter Partes* Review
Patent No. 10,069,945

XI. CONCLUSION

For the reasons given above, Petitioner requests institution of IPR for the challenged claims based on each of the specified grounds.

Respectfully submitted,

Dated: April 29, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,069,945

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,069,945 contains, as measured by the word-processing system used to prepare this paper, 13,970 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: April 29, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,069,945

CERTIFICATE OF SERVICE

I hereby certify that on April 29, 2021, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,069,945 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on PAIR:

Patrick Caldwell (Reg. No. 44580)
G. Gordon (Reg. No. 64517)
P.O. Box 59655
Dept. SVIPGP
Dallas, TX 75229

A courtesy copy was also sent via electronic mail to the Patent Owner's litigation counsel at the following addresses:

TD-PTAB@devlinlawfirm.com
ddahlgren@devlinlawfirm.com
dlflitparas@devlinlawfirm.com
1526 Gilpin Ave.
Wilmington, DE 19806

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)

EXHIBIT 8

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

JENAM TECH, LLC,
Patent Owner

Patent No. 10,069,945

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,069,945**

Petition for *Inter Partes* Review
Patent No. 10,069,945

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	1
III.	PAYMENT OF FEES	3
IV.	GROUND FOR STANDING.....	3
V.	PRECISE RELIEF REQUESTED	3
A.	Claims for Which Review Is Requested	3
B.	Statutory Grounds of Challenge.....	3
VI.	LEVEL OF ORDINARY SKILL	5
VII.	OVERVIEW OF THE '945 PATENT	5
VIII.	CLAIM CONSTRUCTION	8
IX.	DETAILED EXPLANATION OF GROUNDS.....	11
A.	Ground 1: The Combination of <i>Wookey</i> and <i>Berg</i> Render Obvious Claims 7–8	11
1.	Claim 1	11
2.	Claim 7	50
3.	Claim 8	52
B.	Ground 2: The Combination of <i>Wookey</i> , <i>Berg</i> , and <i>Eggert</i> Renders Obvious Claims 11–12	57
1.	Claim 11	57
2.	Claim 12	65
X.	<i>EGGERT</i> IS A PRINTED PUBLICATION.....	66
XI.	DISCRETIONARY DENIAL IS NOT APPROPRIATE	67

Petition for *Inter Partes* Review
Patent No. 10,069,945

A.	<i>Eggert</i> is Not Cumulative.....	67
B.	The Related Litigation Provides No Basis For Discretionary Denial	70
XII.	CONCLUSION.....	78

Petition for *Inter Partes* Review
Patent No. 10,069,945

LIST OF EXHIBITS

EX1001	U.S. Patent No. 10,069,945
EX1002	Declaration of Bill Lin, Ph.D.
EX1003	Curriculum Vitae of Bill Lin, Ph.D.
EX1004	Prosecution History of U.S. Patent No. 10,069,945
EX1005	U.S. Pre-Grant Publication No. 2007/0171921 to Wookey <i>et al.</i>
EX1006	L. Eggert, TCP Abort Timeout Option, draft-eggert-tcm-tcp-abort-timeout-option-00, Network Working Group, Internet-Draft (April 14, 2004) (“ <i>Eggert</i> ”)
EX1007	U.S. Patent No. 6,674,713 to Berg <i>et al.</i>
EX1008	DARPA RFC 793 TRANSMISSION CONTROL PROTOCOL
EX1009	U.S. Patent No. 6,584,546 to Kavipurapu
EX1010	U.S. Patent No. 9,923,995
EX1011	Bova <i>et al.</i> , RELIABLE UDP PROTOCOL <draft-ietf-sigtran-reliable-udp-00.txt> 25 February 1999
EX1012	U.S. Pre-Grant Publication No. 2004/0093376 to De Boor <i>et al.</i>
EX1013	U.S. Patent No. 7,535,913 to Minami <i>et al.</i>
EX1014	U.S. Pre-Grant Publication No. 2005/0204013 to Raghunath <i>et al.</i>
EX1015	U.S. Pre-Grant Publication No. 2007/0005804 to Rideout
EX1016	U.S. Pre-Grant Publication No. 2004/0098748 to Bo <i>et al.</i>
EX1017	<i>Jenam Tech, LLC’s Second Amended Set of Infringement Contentions</i> regarding U.S. Patent No. 10,069,945 (March 17, 2021)
EX1018	IETF RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1
EX1019	U.S. Patent No. 7,636,805 to Rosenberg

Petition for *Inter Partes* Review
Patent No. 10,069,945

EX1020	U.S. Patent No. 6,212,175 to Harsch
EX1021	U.S. Patent No. 8,259,716 to Diab
EX1022	U.S. Patent No. 6,665,727 to Hayden
EX1023	U.S. Patent No. 6,981,048 to Abdolbaghian <i>et al.</i>
EX1024	<i>Jenam Tech., LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Joint Submission of Proposed Amended Scheduling Order, ECF No. 57 (W.D. Tex. Mar. 4, 2021)
EX1025	Fourteenth Supplemental Order Regarding Court Operations Under the Exigent Circumstances Created By the COVID-19 Pandemic, The United States District Court for the Western District of Texas (March 17, 2021)
EX1026	<i>Digital Retail Apps, Inc. v. H-E-B, LP</i> , Case No. 6:19-cv-00167, Joint Stipulation and Order Postponing Trial, ECF No. 182 (W.D. Tex. Jan. 13, 2021)
EX1027	<i>Jenam Tech, LLC's</i> Preliminary Narrowing of Claims (October 20, 2020)
EX1028	<i>Jenam Tech, LLC's</i> Correspondence Regarding Narrowing of Claims (December 4, 2020)
EX1029	Judge Alan D. Albright's Case Statistics By Year (Retrieved from DocketNavigator on March 9, 2021)
EX1030	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Order Granting Motion to Stay Case, ECF No. 58 (W.D. Tex. Mar. 10, 2021)
EX1031	Declaration of Alexa Morris for Eggert
EX1032	U.S. Patent No. 7,464,326 to Kawai
EX1033	IETF RFC 1866 Hypertext Markup Language – 2.0
EX1034	IETF RFC 5482

Petition for *Inter Partes* Review
Patent No. 10,069,945

EX1035	<i>Jenam Tech, LLC's First Set of Infringement Contentions</i> regarding U.S. Patent No. 10,069,945 (August 21, 2020)
--------	--

Petition for *Inter Partes* Review
Patent No. 10,069,945

I. INTRODUCTION

Google LLC (“Petitioner”) requests *inter partes* review (“IPR”) of claims 7–8 and 11–12 (“the challenged claims”) of U.S. Patent No. 10,069,945 (“the ’945 patent”) (EX1001) assigned to Jenam Tech, LLC (“Patent Owner” or “PO”). For the reasons below, the challenged claims should be found unpatentable and canceled.

II. MANDATORY NOTICES

Real Parties-in-Interest: Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies the following as the real parties-in-interest: Google LLC.¹

Related Matters: The ’945 patent is asserted in the following civil actions: *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.); and *Jenam Tech, LLC v. Samsung Group*, Case No. 4:20-cv-00279 (E.D. Tex.) (voluntarily dismissed).

The ’945 patent claims priority to U.S. Patent Application No. 12/714,454 filed February 27, 2010. (EX1001, Cover.) U.S. Patent No. 9,923,995 (“the ’995 patent”) also claims priority to U.S. Patent Application No. 12/714,454. (EX1010,

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

Petition for *Inter Partes* Review
Patent No. 10,069,945

Cover.) The '995 patent is subject to the following instituted IPRs: *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845 ("Google -845 IPR"); and *Unified Patents, LLC v. Jenam Tech, LLC*, IPR2020-00742 ("Unified -742 IPR").

Petitioner is also concurrently filing another IPR petition challenging the '945 patent and IPR petitions challenging U.S. Patent No. 10,306,026 which is also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.).

Petitioner has also filed the following IPR petitions challenging patents related to the '995 and '945 patents, and which are also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.):

- IPR2021-00627 (U.S. Patent No. 10,375,215, "the '215 patent");
- IPR2021-00628 (U.S. Patent No. 10,075,564, "the '564 patent");
- IPR2021-00629 (the '564 patent); and
- IPR2021-00630 (U.S. Patent No. 10,075,565, "the '565 patent").

Counsel and Service Information: Lead counsel: Naveen Modi (Reg. No. 46,224). Backup counsel: (1) Joseph E. Palys (Reg. No. 46,508), (2) Quadeer A. Ahmed (Reg. No. 60,835), (3) Jason Heidemann (Reg. No. 77,880).

Service information is Paul Hastings LLP, 2050 M Street NW, Washington, DC 20036, Tel.: 202.551.1700, Fax: 202.551.1705, email: PH-Google-Jenam-IPR@paulhastings.com. Petitioner consents to electronic service.

III. PAYMENT OF FEES

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-2613.

IV. GROUNDS FOR STANDING

Petitioner certifies that the '945 patent is available for review, and Petitioner is not barred/estopped from requesting review on the grounds herein.

V. PRECISE RELIEF REQUESTED

A. Claims for Which Review Is Requested

Petitioner requests review and cancellation of claims 7–8 and 11–12 as unpatentable based on the following ground.

B. Statutory Grounds of Challenge

Ground 1: Claims 7–8 are unpatentable under pre-AIA 35 U.S.C. § 103(a) over U.S. Publication No. 2007/0171921 to Wookey *et al.* (“*Wookey*”) (EX1005) in view of U.S. Patent No. 6,674,713 to Berg *et al.* (“*Berg*”) (EX1007); and

Ground 2: Claims 11–12 are unpatentable under pre-AIA 35 U.S.C. § 103(a) over *Wookey* (EX1005) in view of *Berg* (EX1007) and further in view of Eggert, TCP Abort Timeout Option (April 14, 2004) (“*Eggert*”) (EX1006).²

² Other references identified herein are provided to show the state of the art at the time of the alleged invention of the '945 patent.

Petition for *Inter Partes* Review
Patent No. 10,069,945

The '945 patent issued from an application filed March 7, 2018, which claims priority through a number of applications back to an application filed February 27, 2010. (EX1001, 1:8–28.) Petitioner assumes for this proceeding only, without conceding, that the earliest effective filing date of the '945 patent is February 27, 2010.

Wookey was published on July 26, 2007, from an application filed November 14, 2006. (EX1005, Cover.) *Berg* was issued on January 6, 2004, from an application filed February 23, 1999. (EX1007, Cover.) Thus, *Wookey* and *Berg* qualify as prior art at least under 35 U.S.C. § 102(b) (pre-AIA).

Eggert, an Internet Engineering Task Force (IETF) Internet-Draft (or “ID”) working document, was published on April 14, 2004. (EX1006, 1; *see also infra* Section X.) As confirmed by the declaration of Alexa Morris, Managing Director of IETF, *Eggert* was published, disseminated, and reasonably available to the public by April 15, 2004. (EX1031 ¶¶9–10; *see also infra* Section X.) Thus, *Eggert* qualifies as prior art at least under 35 U.S.C. § 102(b) (pre-AIA).

The above references were not considered during prosecution of the '945 patent (EX1001, Cover (“References Cited”); *see also generally* EX1004) and are not the same or substantially like any art previously presented to the Office. (*See infra* Section XI.A.)

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art at the time of the alleged invention of the '945 patent ("POSITA") would have had an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking. (EX1002 ¶¶16–18.)³ More education can supplement practical experience and vice versa. (*Id.*)

VII. OVERVIEW OF THE '945 PATENT

The '945 patent is directed to networking, and to the sharing of information for detecting an idle TCP connection. (EX1001, 2:26–28, 8:33–55; EX1002 ¶¶64–72.) Figure 7 illustrates such a process.

³ Petitioner submits the declaration of Dr. Bill Lin (EX1002), an expert in the field of the '945 patent. (EX1002 ¶¶3–15; EX1003.)

Petition for *Inter Partes* Review
 Patent No. 10,069,945

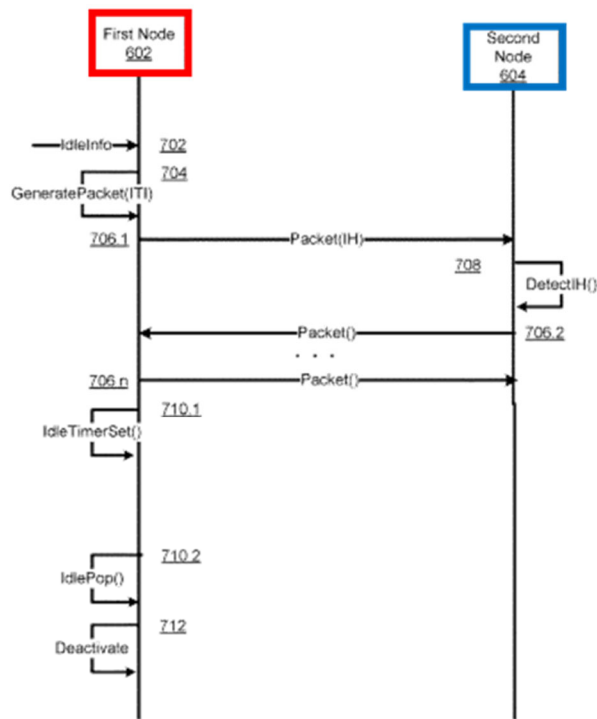


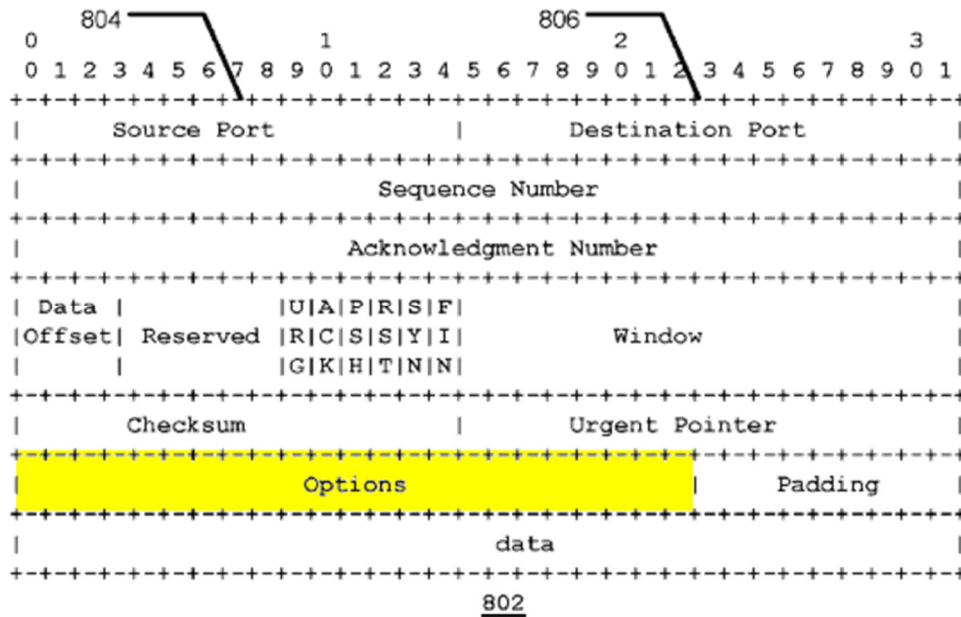
Fig. 7

(EX1001, FIG. 7 (annotated); EX1002 ¶64.)

First, the **first node 602** receives a message 702 that identifies idle information representing a duration for an idle time period (“ITP”). (EX1001, 11:59–67.) Message 702 may take various forms, such as “a message received via a network.” (*Id.*, 11:62–67.) The idle information “may include and/or identify a duration of time” for detecting an ITP. (*Id.*, 12:40–44.) The duration “may be specified according to various measures of time[,] including seconds.” (*Id.*; EX1002 ¶65.)

Petition for *Inter Partes* Review
 Patent No. 10,069,945

Next, the **first node** “generat[es] a TCP packet including an ITP header based on received idle information.” (EX1001, 16:6–9.) The **TCP options field** of a TCP packet may store the ITP header. (*Id.*, FIG. 8) Figure 8 below, “adapted from RFC 793,” illustrates this configuration. (*Id.*, FIG. 8, 6:31–33, 15:48–56.)



(*Id.*, FIG. 8 (cropped-annotated); EX1002 ¶66.)

The TCP options field contains a KIND sub-field that identifies the type of option presented, a length sub-field that specifies the length of the option field, and an **ITP Data** sub-field containing data. (EX1001, FIG. 8.)

Petition for *Inter Partes* Review
 Patent No. 10,069,945

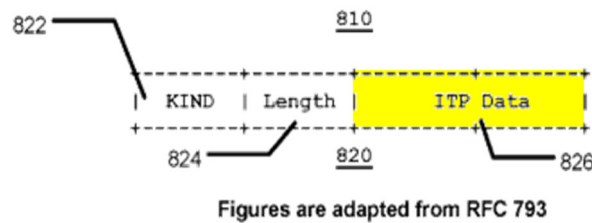


Fig. 8

(*Id.* (cropped-annotated); EX1002 ¶¶66.)

The ITP header is exchanged during the three-way handshake (EX1001, 14:53–67.) For example, the **first node** transmits a message 706.1, i.e., a TCP packet including an ITP header (IH) containing **ITP information**, to the **second node 604**. (*Id.*, 16:31–33, FIG. 7.) Message 708 exemplifies the **second node**'s detection of the ITP header in the received TCP packet. (*Id.*, 21:52–55; EX1002 ¶¶67; *see also* EX1002 ¶¶68–71.)

All the limitations in the challenged claims were known in the prior art and obvious. (*See* Section IX; EX1002 ¶¶72; *see also* EX1002 ¶¶19–63 (technology background), citing EXS. 1008–1009, 1011–1016, 1018–1023.)

VIII. CLAIM CONSTRUCTION

Under the applicable standard in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), claim terms are typically given their ordinary and customary meanings as understood by a POSITA at the time of the invention based on the claim

Petition for *Inter Partes* Review
Patent No. 10,069,945

language, specification, and the prosecution history of record. *Phillips*, 415 F.3d at 1313; *see also id.* at 1312–16. The Board, however, only construes the claims when necessary to resolve the controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper No. 11 at 16 (Aug. 14, 2015) (citation omitted). Petitioner believes no express constructions of any claim terms are necessary to assess whether the prior art reads on the challenged claims. (EX1002 ¶73.)

Claim 1, from which challenged claims 7–8 and 11–12 depend, recites the term “a second protocol that is separate from the TCP.” (EX1001, 24:38–39.) The ’945 patent mentions “separate from the TCP” only in the “Summary” section, which the ’945 patent explains is “not an extensive overview of the disclosure and it does not identify key/critical elements of the invention or delineate the scope of the invention.” (*Id.*, 2:32–39, 2:63, 3:28–29.) As discussed below, *Berg*’s RUDP protocol is separate from the TCP protocol. (*See, e.g., infra* Section IX.A.1.f.)

Moreover, the Board preliminarily found *Berg* discloses a non-TCP protocol. *See, e.g.,* Unified -742 IPR, Paper 11 (Institution Decision) at 10–11. Thus, because Petitioner relies on *Berg*’s similar disclosures to satisfy the “protocol that is separate from the TCP” terms herein (*see infra* Section IX.A.1.f.), and given the ’945 patent offers no evidence requiring a special meaning of this term and the prior art discloses

Petition for *Inter Partes* Review
Patent No. 10,069,945

this feature under any reasonable interpretation, construction of this term is unnecessary.⁴

⁴ Petitioner reserves all rights to raise claim construction and other arguments, including challenges under 35 U.S.C. §§ 101 or 112, in district court as relevant to those proceedings. *See, e.g., Target Corp. v. Proxicom Wireless, LLC*, IPR2020-00904, Paper 11 at 11–13 (November 10, 2020). A comparison of the claims to any accused products in litigation may raise controversies that are not presented here given the similarities between the references and the patent.

IX. DETAILED EXPLANATION OF GROUNDS**A. Ground 1: The Combination of *Wookey* and *Berg* Render Obvious Claims 7–8**

Petitioner first addresses claim 1 as claims 7-8 depend from claim 1.

1. Claim 1**a) A computer-implemented method, comprising:**

To the extent that the preamble of claim 1 is limiting, *Wookey* discloses the limitations therein. (EX1002 ¶¶104–111.)

Wookey discloses a computer-implemented method, as claimed. For example, *Wookey* teaches “a *method* for making a hypermedium page interactive,” using a client machine and one or more remote machines. (EX1005, Abstract; *id.* ¶[0002]; EX1002 ¶106; *see also* EX1002 ¶¶74–83 (*Wookey* overview).)⁵

Figures 1, 2C, and 3D below show an exemplary environment in which a **client machine 10**, **remote machine 30**, and **remote machine 30'** are employed to carry out the method. (*See also* EX1005 ¶¶[0020], [0024], [0026], [0192]–[0196], [0207].) *Wookey*'s client and remote machines are “typical computers” (*id.* ¶[0021], FIGS. 1A–1B) including personal computers and wireless devices (e.g., mobile telephones or other portable telecommunication devices) (*id.* ¶¶[0171]–[0173]).

⁵ Emphasis is added unless otherwise stated.

Petition for *Inter Partes* Review
 Patent No. 10,069,945

FIG. 1

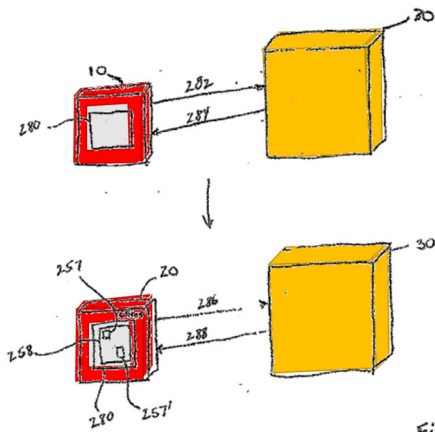
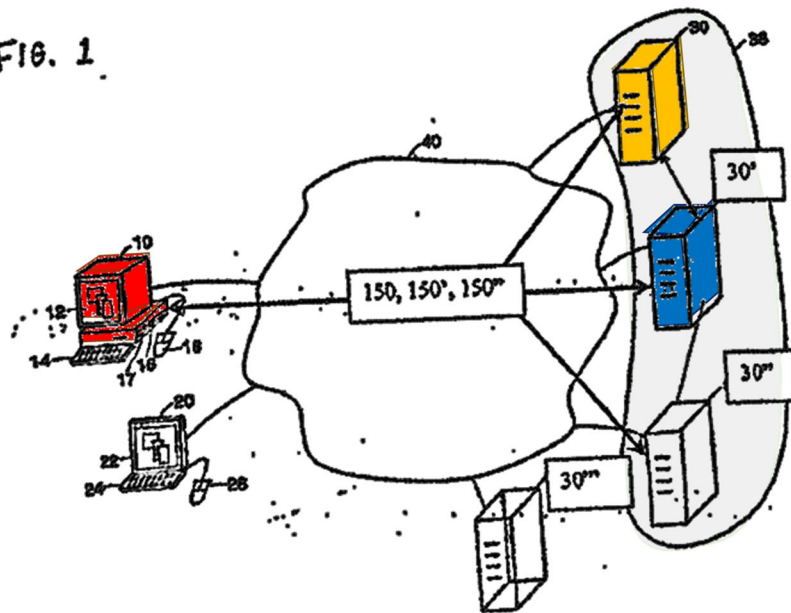


Fig. 2C

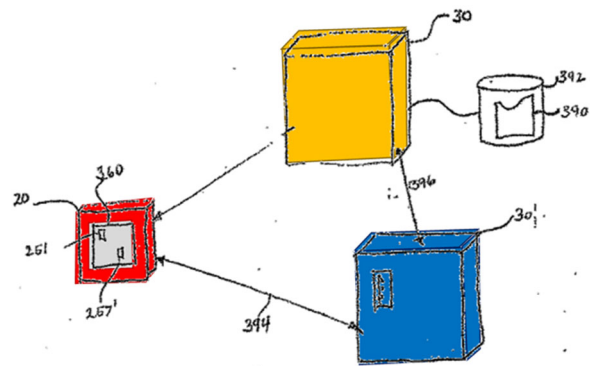


Fig. 3D

(EX1005, FIGS. 1, 2C, 3D (each annotated); EX1002 ¶¶107–108; *see also* EX1002 ¶¶109–110.)

Thus, *Wookey* discloses “[a] computer-implemented method,” as claimed.
 (See *infra* Sections IX.A.1.b–j; EX1002 ¶111.)

b) causing access to be provided to a server computer including:

- (1) a non-transitory memory storing a network application, and
- (2) one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application to operate in accordance with a first protocol including a transmission control protocol (TCP);

Wookey discloses this limitation. (EX1002 ¶¶112–129.) For example, *Wookey* discloses that its above-noted computer-implemented method causes access to be provided to a **remote machine 30** (“server computer”). (EX1005 ¶[0160] (“the remote machines 30 ... are provided as ... computer servers”). The access is caused to be provided, for example, by allowing **client machine 10** to gain access to resources provided by **remote machine 30**, as discussed further below with respect to claim limitations 1.d–f. (EX1002 ¶113; *see infra* Sections IX.A.1.d–f.)

Remote machine 30 is configured with typical computer architecture.

FIG. 1A

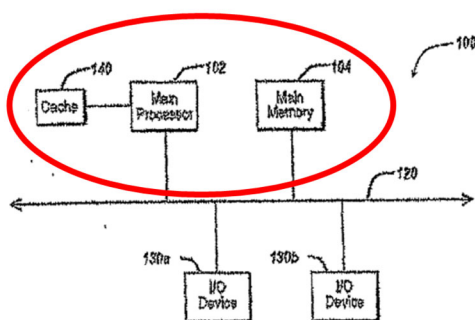
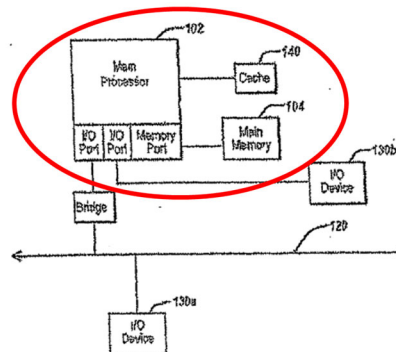


FIG. 1B



Petition for *Inter Partes* Review
Patent No. 10,069,945

(EX1005, FIGS. 1A and 1B (annotated), ¶¶[0016], [0021]; EX1002 ¶114.)

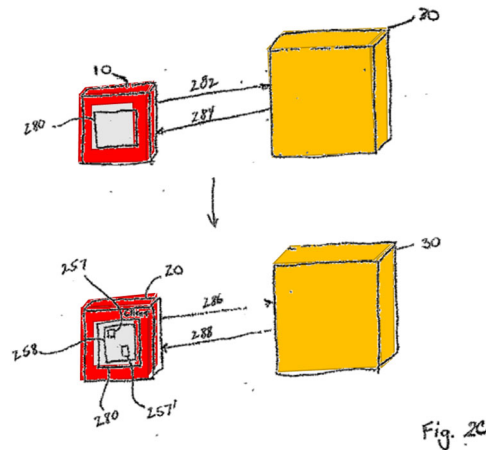
In this architecture, **remote machine 30** includes a main memory unit 104 and cache memory 140 (individually or collectively, the claimed “non-transitory memory”) which are in communication with a main processor 102 (“one or more processors”). (EX1005 ¶¶[0161]–[0165]; EX1002 ¶114.)

Main memory 104 and cache memory 140 store instructions (e.g., software). For instance, “*instructions [are] fetched from the main memory unit 104*” by main processor 102. (EX1005 ¶¶[0162]–[0163]; *id.* ¶[0158], FIGS. 1A–1B.) Also, processor 102 communicates with cache memory 140, which stores data and instructions accessible to the processor. (*Id.* ¶[0165]; EX1002 ¶¶115–117 (*citing* EX1009, 1:15–39).)

Main memory 104 and cache memory 140 store a network application, as claimed, because the above instructions (e.g., software) enable **remote machine 30** to act as a web server (“network application”). A web server is an example of a network application because it includes software that runs on one computer (e.g., **remote machine 30**) which provides communication to another application (e.g., web browser 280) running on another computer (e.g., **client machine 10**). (EX1005 ¶¶[0179], [0207], [0213].) *Wookey* describes an arrangement in Figure 2C where **remote machine 30** “acts as a web server” at least because it communicates with

Petition for *Inter Partes* Review
 Patent No. 10,069,945

client machine 10 to authenticate, and provide a list of resources to, **client machine 10**. (EX1005 ¶[0207], [0213].)



(*Id.*, FIG. 2C (annotated); EX1002 ¶118.)

Main processor 102 executes instructions for executing **remote machine 30** as a web server, such that it operates in accordance with a first protocol including a transmission control protocol (TCP), as claimed. For example, **client machine 10** executes a web browser application 280 (EX1005 ¶[0192]) that “transmits a request 282 to access a Uniform Resource Locator (URL) address corresponding to an HTML [HyperText Markup Language] page residing on” **remote machine 30** (*id.* ¶[0193]). (*Id.* ¶¶[0024], [0207], [0213], [0279].) The request is received and handled by **remote machine 30**. (*E.g., id.* ¶¶[0193]–[0196].) **Remote machine 30** includes functionality for authenticating (*id.* ¶[0194]) and transmitting an HTML

Petition for *Inter Partes* Review
Patent No. 10,069,945

page to **client machine 10** (*id.* ¶[0196]). (*See also id.* ¶¶[0179], [0207], [0213]–[0215], [0803]–[0808] (web server functionality), FIGS. 2C, 3D; EX1002 ¶¶119–123.)

The communication link 150 between **client machine 10** and **remote machine 30** uses the TCP. (EX1005 ¶¶[0154]–[0156], [0174], [0731]–[0732].) Thus, in the communication between **client machine 10** and **remote machine 30** for Figure 2C, **remote machine 30** acts as a web server that operates in accordance with a first protocol including TCP, as claimed. (EX1002 ¶¶124–126.)

Accordingly, *Wookey's* **remote machine 30**, which can perform processes over a TCP connection to handle requests from **client machine 10**, executes instructions via main processor 102 such that its web server operates in accordance with a first protocol including a TCP. (*Id.* ¶127.) Given processor 102 performs the processing functionalities of **remote machine 30** (EX1005 ¶[0021], FIGS. 1A–1B), processor 102 would thus execute the instructions such that **remote machine 30** acts as a web server in accordance with a first protocol including a TCP, to process, e.g., the received request 282 from **client machine 10**. (EX1002 ¶¶128–129.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

- c) **causing a TCP connection to be established with a client computer, by:**
 - communicating a segment including at least one first synchronize bit;**
 - communicating a first acknowledgement of the segment, and at least one second synchronize bit; and**
 - communicating a second acknowledgement;**

Wookey discloses this limitation. (EX1002 ¶¶130–136.) As discussed, **remote machine 30** (“server computer”) communicates with **client machine 10** (“client computer”) over a TCP connection. (*Supra* Section IX.A.1.b.) As explained below, when the machines are communicating over a TCP connection, *Wookey*’s method would have necessarily caused a TCP connection to be established between **remote machine 30** and **client machine 10** by performing a three-way handshake, which would involve the three “communicating” processes recited in this limitation. (EX1002 ¶131.)

A three-way handshake method is necessarily performed to establish a TCP connection. (*Id.* ¶132; *see also id.* ¶¶46–47.) For example, RFC 793 (EX1008), the TCP specification, provides that a three-way handshake is a necessary procedure used to establish a TCP connection. (EX1008, 32, 34; *id.*, 31, 34–37.) The three-way handshake was known to involve transmitting three messages to negotiate and start a TCP-based session between two devices. (EX1002 ¶133; EX1008, 31–32;

Petition for *Inter Partes* Review
Patent No. 10,069,945

34–37.) The ’945 patent acknowledges such understandings, citing RFC 793. (EX1001, 14:53–67.)

The *first* communicated message (step 1) includes a segment with a control bit for synchronization (SYN bit) and thus discloses “communicat[ing] a segment including at least one first synchronize bit,” as claimed. (EX1008, 31; EX1001, 14:53–61.) The *second* communicated message (steps 2 and 3 combined) includes an acknowledgment (ACK) to the segment with the SYN bit and a second SYN bit, and thus discloses “communicat[ing] a first acknowledgement of the segment, and at least one second synchronize bit,” as claimed. (EX1008, 31–32; EX1001, 14:62–63, 14:65–67.) The *third* communicated message (step 4) includes a second ACK to the second SYN bit and thus discloses “communicat[ing] a second acknowledgement,” as claimed. (EX1008, 31–32; EX1001, 14:64–67; EX1002 ¶134.)

Thus, to establish a TCP connection with **client machine 10**, **remote machine 30** necessarily performs a three-way TCP handshake that includes the same features recited in this limitation because this handshake was/is a *required* procedure for establishing a TCP connection. (EX1002 ¶¶135–136.) Indeed, PO relies on the

Petition for *Inter Partes* Review
Patent No. 10,069,945

same three-way handshake disclosures in RFC 793 for its allegations regarding this limitation. (EX1017, 5–6.)⁶

d) **causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer;**⁷

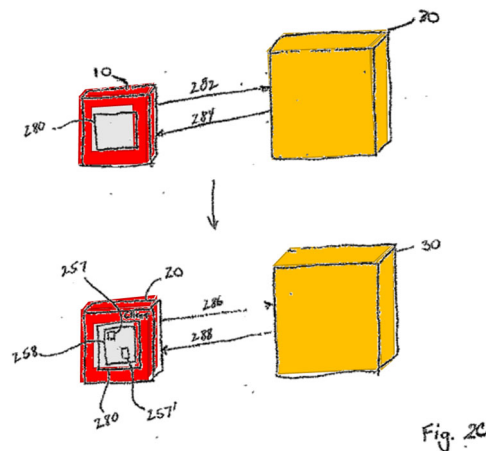
Wookey discloses this limitation. (EX1002 ¶¶137–144.) For example, *Wookey* discloses that upon **client machine 10** transmitting a request 282 to **remote machine 30** for accessing a URL address corresponding to an HTML page residing on **remote machine 30**, **remote machine 30** is caused to communicate an HTML page 284 which is an authentication page (“first data”) to **client machine 10** utilizing

⁶ Petitioner’s references to PO’s infringement allegations are not indicative of any concession that any accused instrumentality infringes any claim limitation as alleged by PO.

⁷ The ’945 patent specification does not describe the claimed “causing first data ...,” “causing ... to permit second data ...,” and “code” features as recited in limitations 1.d–f and claims 7–8. (EX1002 n.10, n.12, n.13.) Therefore, while Petitioner addresses these limitations under Ground 1–2, Petitioner does not concede that the ’945 patent provides adequate disclosures of such features.

Petition for *Inter Partes* Review
 Patent No. 10,069,945

the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user the **client machine 10**. (EX1005 ¶¶[0192]–[0194], FIG. 2C.)



(*Id.*, FIG. 2C (annotated); EX1002 ¶138.)

In particular, **client machine 10** executes a web browser application 280 that “transmits a request 282 to access a Uniform Resource Locator (URL) address corresponding to an HTML page residing on remote machine [30]”. (EX1005 ¶¶[0192]–[0193].) **Remote machine 30** is then caused to communicate an HTML page 284 to **client machine 10**. (*Id.* ¶[0193]; EX1002 ¶139.)

HTML page 284 is “an authentication page [284] that seeks to identify the client machine 10 or the user of the client machine 10” (EX1005 ¶[0193]) and includes data such as tags and associated information defining the layout of a

Petition for *Inter Partes* Review
Patent No. 10,069,945

webpage and what is to be presented on the webpage. (EX1002 ¶140; *see also* EX1005 ¶¶[0196], [0207]–[0214] (discussing an exemplary webpage that **remote machine 30** transmits to **client machine 10**), [0754]–[0755] (same).)

The understanding that an HTML page including data meets this limitation is consistent with PO’s interpretation of this limitation. (EX1017, 6 (alleging “causing first data to be communicated from the server computer to the client computer (e.g., device that receives the HTML pages, etc.).”).)

“The authentication page [284] allows the client machine 10 to transmit user credentials, via the web browser 280, to the remote machine 30 for authentication.” (EX1005 ¶[0194], FIG. 2C.) Thus, the HTML authentication page 284 is “for being presented to a user of the client computer,” as claimed, because **client machine 10**’s user has to review the authentication request presented on web browser 280 and provide credentials. (*Id.*; EX1002 ¶141.)

Moreover, the HTML authentication page 284 is communicated from **remote machine 30** to **client machine 10** “utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP),” as claimed. As discussed above, the communication link 150 between **client machine 10** and **remote machine 30** is a TCP connection operating in accordance with the TCP protocol. (*See supra* Sections IX.A.1.b–c.) *Wookey* further discloses that this

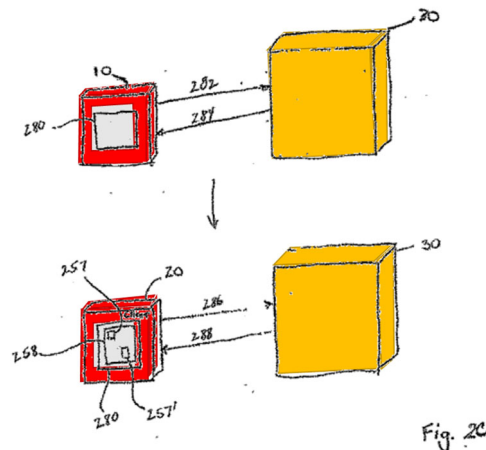
Petition for *Inter Partes* Review
Patent No. 10,069,945

communication link 150 uses HTTP at the application layer. (EX1005 ¶[0155] (“[t]he communications link 150 may use ... any application layer protocol, such as the Hypertext Transfer Protocol (HTTP)”); *see also id.*, ¶¶[0159], [0551]–[0556] (describing that the server agent and client agent can communicate using the HTTP protocol), [0899]; EX1002 ¶142.) Thus, *Wookey*’s method causes **remote machine 30** to act as a web server to communicate HTML pages to **client machine 10** over a TCP connection in accordance with TCP and HTTP, for being presented to the user of **client machine 10**. (EX1002 ¶¶143–144.)

Petition for *Inter Partes* Review
 Patent No. 10,069,945

e) causing the server computer to permit second data, from the user of the client computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP);⁸ and

Wookey discloses this limitation. (EX1002 ¶¶145–149.) For example, *Wookey*'s method causes **remote machine 30** to allow (“permit”) user credentials (“second data”), from the user of **client machine 10**, to be received at **remote machine 30** from **client machine 10** utilizing the TCP connection in accordance with TCP and HTTP. (EX1005 ¶¶[0192]–[0196], FIG. 2C.)



(*Id.*, FIG. 2C (annotated); EX1002 ¶146.)

⁸ See *supra* n.7.

Petition for *Inter Partes* Review
Patent No. 10,069,945

As discussed above, the communication link 150 between **client machine 10** and **remote machine 30** is a TCP connection operating in accordance with the TCP protocol and HTTP. (*See supra* Sections IX.A.1.b–d.) In response to receiving the HTML authentication page 284, **client machine 10** transmits user credentials (e.g., via communication 286) to **remote machine 30** for authentication over this TCP connection. (*Id.*; EX1005 ¶¶[0193]–[0196], FIG. 2C; EX1002 ¶147.)

In particular, *Wookey* discloses that “[t]he authentication page allows the client machine 10 to transmit user credentials, via the web browser 280, to the remote machine 30 for authentication” and these “[t]ransmitted user credentials are verified” by one of the remote machines. (EX1005 ¶[0194].) *Wookey* similarly discloses that “an access control decision is made based on *received information* about the user resources available to the user of the client system,” where “[t]he remote machine 30 may verify the *user credentials received* from the client machine 10” or “the remote machine 30 may pass the *user credentials* to another remote machine for authentication.” (*Id.* ¶[0195].) The received information regarding the user’s credentials is “data,” as claimed, because it is used to review and make decisions regarding the level of access control to resources that can be provided to **client machine 10**. (EX1002 ¶¶148–149.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

f) causing access to be provided, to the client computer, to code that causes the client computer to operate in accordance with a second protocol that is separate from the TCP, in order to establish a second protocol connection with another server computer, by:⁹

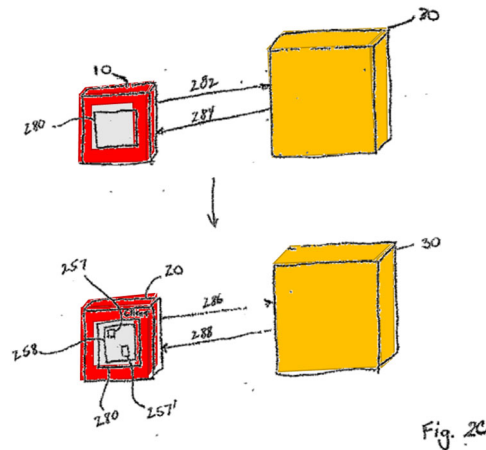
The *Wookey-Berg* combination discloses and/or suggests this limitation. (EX1002 ¶¶150–188.) For example, *Wookey*’s method causes access to be provided, to **client machine 10**, to an HTML page 288 containing encoded URLs, in order to establish a second protocol connection with **remote machine 30’** (“another server computer”) that is different from **remote machine 30**. As discussed below, it would have been obvious in light of *Berg*’s teachings to configure *Wookey*’s HTML page 288 containing the encoded URLs such that the HTML page 288 containing the encoded URLs would cause **client machine 10** to operate in accordance with an RUDP protocol (“second protocol that is separate from the TCP”), in order to establish an RUDP connection (“second protocol connection”) with **remote machine 30’**. (*Id.* ¶151.)

Referring to Figure 2C below, once **client machine 10** is authenticated (e.g., based on the provided user credentials), “remote machine [30] prepares and transmits to the client machine 10 an HTML page 288 that includes a Resource

⁹ See *supra* n.7.

Petition for *Inter Partes* Review
 Patent No. 10,069,945

Neighborhood window 258 in which appears graphical icons 257, 257' representing resources to which the client machine 10 has access.” (EX1005 ¶[0196].)

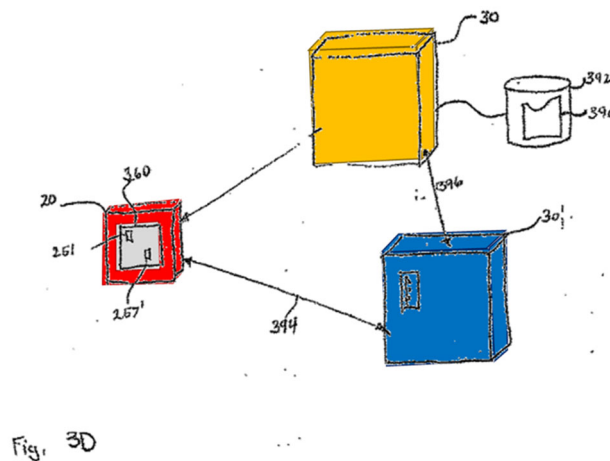


(*Id.*, FIG. 2C (annotated); EX1002 ¶152.)

The HTML page 288 includes encoded URLs associated with icons (e.g., 257, 257') on the HTML page 288, and both (1) the encoded URLs included in the HTML page or (2) the HTML page including the encoded URLs disclose the claimed “code.” (EX1002 ¶153.) For example, “[e]ach icon 257, 257' is associated with ***an encoded URL that specifies: the location of the resource ...; a launch command associated with the resource; and a template identifying how the results of accessing the resource should be displayed.***” (EX1005 ¶[0216].) Each encoded URL also “***contains the information necessary for the client to create a connection to the remote machine hosting the resource.***” (*Id.*) Thus, when a user *clicks* an

Petition for *Inter Partes* Review
 Patent No. 10,069,945

icon 257, 257' corresponding to a resource from the displayed HTML page 288, **client machine 10** uses the encoded URL associated with the clicked icon to initiate the process for establishing a second connection (e.g., connection 394 in Figure 3D) with **remote machine 30'**. (EX1002 ¶153; *see also* EX1005 ¶¶[0026], [0213]–[0216], FIG. 3D.) Accordingly, the encoded URLs are the claimed “code” because such encoded URLs are instructions that, when used by **client machine 10**, command **client machine 10** to access resources from one or more locations (e.g., one or more remote machines). (EX1002 ¶154.)



(EX1005, FIG. 3D (annotated); EX1002 ¶154.)

The above understanding is consistent with how the '945 patent claims describe its “code” as an “instruction for connecting to the another server.” (EX1001, cl. 41 (28:20-22).) *Wookey*’s encoded URLs similarly include an

Petition for *Inter Partes* Review
Patent No. 10,069,945

“instruction,” such as the launch command to connect to **remote machine 30’**. (EX1002 ¶155.)

Moreover, the entire HTML page 288 (including the encoded URLs and icons) also discloses the claimed “code that causes the client computer,” because an HTML page was known to comprise HTML code used to define, e.g., what is displayed on the screen of the client node. (EX1002 ¶156 (citing EX1012).) The HTML code used to display information relating to page 288 would include the icons associated with the encoded URLs, and when **client machine 10** uses one of the encoded URLs, it causes **client machine 10** to operate in accordance with a second protocol separate from the TCP to set up a second protocol connection with another server (e.g., **remote machine 30’**). PO similarly relates the claimed code to “HTML pages” in its infringement assertions. (EX1017, 8 (“causing access to be provided, to the client computer (e.g., device that receives the HTML pages, etc.), to code.”).)

Wookey’s method further discloses “causing access to be provided,” to **client machine 10**, to this “code” because it discloses **remote machine 30** causing access to be provided, to **client machine 10**, to HTML page 288 including the encoded URLs, so **client machine 10** can request access to a resource represented by the icons 257/257’ on the HTML page 288. (EX1005 ¶[0196]; EX1002 ¶157.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

Thus, *Wookey* discloses, “causing access to be provided, to the client computer, to code that causes the client computer to operate” such that a process for establishing a second connection (e.g., connection 394 in Figure 3D) with **remote machine 30’** is initiated.¹⁰ (EX1002 ¶158.)

Regarding the connection established between machines **10** and **30’**, **client machine 10** includes an “HTTP client agent,” such as the web browser application 280, which “can use any type of protocol.” (EX1005 ¶[0159]; *see also id.* ¶¶[0155], [0192]–[0195].) For example, connection 394 can be made using various protocols, including the Virtual Network Computing (VNC) protocol. (*Id.* ¶[0216].) VNC is a desktop-sharing system that allows a computer to remotely access and control another computer. (EX1002 ¶159; EX1014 ¶[0019].) VNC can run over various “industry standard transport protocols, such as TCP/IP, IPX/SPX, NetBEUI.” (EX1005 ¶[0225]; *see also id.* ¶¶[0155], [0215]–[0216].) IPX/SPX and NetBEUI protocols, being non-TCP–based transport protocols, are examples of protocols that are separate from the TCP protocol. (EX1002 ¶¶160–161; EX1013, 1:20–33.)

¹⁰ Petitioner’s analysis regarding the “code” limitation equally applies whether the “code” is the encoded URLs in the HTML page or the HTML page itself (including the encoded URLs). (EX1002 ¶158 n.16.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

Wookey further explains that **client machine 10** may use “a different type of protocol than the one used to send the request to the remote machine 30.” (EX1005 ¶[0732]; *see also id.*, (“the client machine 10 may make a request to the remote machine 30 using the IPX protocol and request the address of the remote machine 30’ as a TCP/IP protocol address”); EX1002 ¶162.)

Wookey also identifies desired characteristics associated at least with the second connection that a POSITA would have considered when choosing a protocol, such as negotiating connection parameters, reducing unintentional session terminations due to an imperfect connection, detecting/handling disconnections like when a mobile device enters an elevator, and ability to specify an inactive time before connection termination. (EX1002 ¶163; EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153].)

Thus, **client machine 10** identifies negotiable parameters for the connection to be established. (EX1002 ¶164.) For example, **client machine 10** and **remote machine 30’** “exchange a set of messages which negotiate the parameters under which communications will occur.” (EX1005 ¶[0774]; *see also id.* ¶¶[0738]–[0739] (describing the **remote machine 30’** listening for connection requests and processing them), [0744], [0772]–[0773].)

Petition for *Inter Partes* Review
Patent No. 10,069,945

So when **client machine 10** uses one of the encoded URLs (e.g., the one pointing to **remote machine 30'**) associated with one of the icons on the HTML page 288, it results in **client machine 10** utilizing, e.g., the web browser application 280, to establish a connection (e.g., connection 394) with **remote machine 30'** in accordance with a transport protocol. (EX1002 ¶¶165–166.)

While *Wookey* discloses various protocols (including protocols that are separate from TCP (e.g., IPX)) may be used to establish the second connection and the desired characteristics of such a connection (EX1005, ¶[0225]), *Wookey* does not expressly provide details regarding establishing and maintaining the second connection or the types of negotiable parameters used with those protocols for the second connection. (EX1002 ¶167.) However, it would have been obvious to a POSITA to consider and implement a protocol following *Wookey's* desired characteristics (e.g., such as those identified above) for the second connection between **client machine 10** and **remote machine 30'**, which may be “a different type of protocol than the one used to send the request to the remote machine 30.” (EX1005 ¶[0732].)

Berg describes such a protocol (a “*protocol that is separate from the TCP*”) that allows nodes to negotiate timeout timers on a per-connection basis to provide a reliable, efficient network connection. (EX1002 ¶167.) Accordingly, based on the

Petition for *Inter Partes* Review
Patent No. 10,069,945

teachings of *Berg*, a POSITA’s knowledge, and guidance from *Wookey*, it would have been obvious to configure *Wookey*’s **client machine 10** such that when the “code” (e.g., the encoded URL(s) in HTML page 288 pointing to a resource on **remote machine 30**’, or the HTML page 288 including such URL(s)) is used, the code causes **client machine 10** to operate per a second protocol such as the one described by *Berg* (which would have been separate from the TCP protocol used to send the request to **remote machine 30**) having at least some characteristics contemplated by *Wookey* for such a second connection between **client machine 10** and **remote machine 30**’. (EX1002 ¶168.)

i. Berg

Berg, like *Wookey*, relates to establishing a reliable connection between a client and server as it states “[t]o provide a reliable backhaul, it is important to be able to maintain multiple IP connections or sessions between the signaling terminal device and the call processing device, so that message[s] can be transmitted even when the *inherently unreliable IP network fails*” and “[t]he system must be configured so that *communications are not interrupted upon network failure*, and so that *communications can resume when the network comes back up*.” (EX1007, 2:4–11; EX1002 ¶¶84–97 (*Berg* overview).) Thus, *Berg* discloses networking features in the same technical field as *Wookey*. (See, e.g., EX1005 ¶¶[0581] (*Wookey*

Petition for *Inter Partes* Review
Patent No. 10,069,945

disclosing that traversal of network segments by **client machine 10** may cause changes in its network address or host name or causes it “to disconnect”), [1135]–[1136] (discussing the need for connections to manage unintentional disconnection between the client and remote machines), [1153] (**client machine 10** losing connection because the user enters an elevator); *id.* ¶¶[0013], [0196], FIGS. 1, 2C, 3D.) A POSITA, therefore, would have had reason to consider *Berg*’s teachings when implementing *Wookey*’s method for establishing a second protocol connection between **remote machine 30’** and **client machine 10**. (EX1002 ¶¶169–170.)

Berg provides a gateway device (acting as a client) connected to a media gateway controller (acting as a server). (EX1007, 3:14–23, 2:55–60.) In *Berg*’s system, both the client and server devices execute a session manager that manages the data network communication session. (*Id.*, 2:54–60, 7:63–8:2, 8:24–31; EX1002 ¶171.) The session manager “operates logically above a *reliable communication layer*”, which “determines when or if a session is connected or failed.” (EX1007, 2:54–60.) *Berg* explains “[a] protocol layer is ‘reliable’ when it guarantees that it will deliver or generate an error message for every message that an application program requests the transport layer using that protocol to transport.” (*Id.*, 8:27–31.) *Berg*’s Figure 2 below describes the layers that computer systems

use for network communications based on the Open Systems Interconnection (OSI) reference model. (*Id.*, 8:3–17, 5:27–55.)

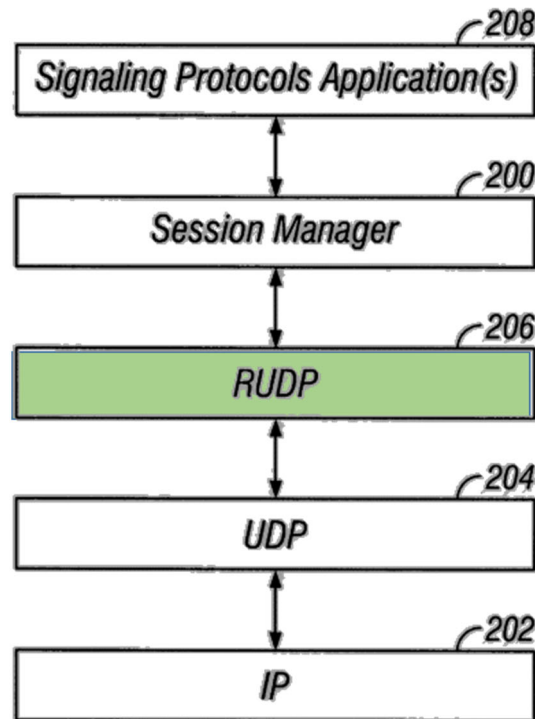


FIG. 2

(EX1007, FIG. 2 (annotated); EX1002 ¶171.)

The **Reliable User Datagram Protocol (RUDP) 206** runs on top of the User Datagram Protocol (UDP) protocol software 204 at a transport layer. (EX1007, 8:10–17, 16:66–17–17; EX1002 ¶172.) The UDP is a non-TCP communications protocol (e.g., a protocol that is separate from the TCP) that is primarily used for establishing low-latency and loss-tolerating connections, especially time-sensitive transmissions, between applications. (EX1002 ¶172; EX1015 ¶[0110]; EX1016

Petition for *Inter Partes* Review
Patent No. 10,069,945

¶[0010].) The RUDP layer 206 allows *Berg*’s system to determine “when or if a session is connected or failed.” (EX1007, 2:54–60.)¹¹ Above the **RUDP layer 206** is the Session Manager 200 and Signaling Software Application(s) 208. (*Id.*, 8:24–27.)

The Session Manager 200 can run above any reliable communication mechanism (e.g., RUDP 206). (*Id.*) A communication layer “is ‘reliable’ when it guarantees that it will deliver or generate an error message for every message that an application program requests the transport layer using that protocol to transport.” (*Id.*, 8:27–31; *see also id.*, 8:11–13; EX1002 ¶173.)

Berg notes that the “TCP/IP has a number of characteristics that make it unsuitable for” some applications. (EX1007, 17:28–30.) In particular, “[m]ost TCP/IP implementations that allow properties like timers to be modified, *do not allow* the modification to be done on a per-connection basis.” (*Id.*, 17:40–42; EX1002 ¶174.)

¹¹ *Berg* uses “RUDP layer 206” and “communication layer” to refer to the same layer in the exemplary OSI model in Figure 2. (EX1002 ¶172 n.19; EX1007, 2:54–60, 8:13, 8:24–28.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

Contrasted to TCP/IP implementations, “RUDP is designed *to allow characteristics of each connection to be individually configured* so that many protocols with different transport requirements can be implemented simultaneously on the same platform.” (EX1007, 17:42–46.) For example, RUDP allows the nodes to negotiate various parameters including the timeout values associated with timers for the retransmission timeout (*id.*, 20:24–29, 22:44–56), acknowledgement timeout (*id.*, 20:29–35, 23:15–24), null segment (*id.*, 20:36–41, 23:45–59), and transfer state (*id.*, 20:42–47, 24:15–25). *Berg* explains that the messages sent to set up the RUDP connection are UDP packets that include an RUDP header. (*Id.*, 17:59–18:20, 19:1–40; EX1002 ¶175.)

The “RUDP is a lightweight protocol layer designed to run on top of UDP [that] can provide reliable in-order delivery” (EX1007, 16:66–67; *see also id.*, 17:9–16) and it “has a *very flexible design* that would make it *suitable for a variety of transport uses*. (*Id.*, 17:13–15; EX1002 ¶176.)

Thus, the RUDP is “a second protocol that is separate from the TCP” (e.g., a non-TCP protocol), and operating in accordance with the RUDP would have caused a client to set up an RUDP connection (“a second protocol connection”) with a server. (EX1002 ¶177.) *See also* Unified -742 IPR, Paper 11 at 10 (Board agreeing that *Berg*’s RUDP is a non-TCP protocol).

ii. Reasons to Combine

Based on *Wookey*'s and *Berg*'s disclosures and a POSITA's knowledge at the time, it would have been obvious to configure and implement *Wookey*'s **client machine 10** (including its web browser application 280) to use a protocol like *Berg*'s RUDP ("a second protocol that is separate from the TCP") in order to set up an RUDP ("second protocol") connection with **remote machine 30'** (see, e.g., EX1005, Figure 3D) in relation to claim limitations 1.f–j and claims 7–8 and 11–12. (EX1002 ¶178.) Such an implementation would have had many advantages and been a predictable and straightforward result achieved by combining well-known technologies using known methods. See *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 416–18 (2007).

Both *Wookey* and *Berg* are directed to establishing a reliable connection between two nodes, where one or both nodes may experience an unintentional disconnection, and thus disclose features in a similar technological field. (EX1002 ¶¶178–179; *supra* Section IX.A.1.f.i.) Thus, a POSITA would have had reason to consider *Berg* when contemplating and implementing the teachings of *Wookey*. (EX1002 ¶¶178–179.) And when collectively considered, such a POSITA would have been motivated to modify *Wookey*'s **client machine 10** to use a protocol like

Petition for *Inter Partes* Review
Patent No. 10,069,945

Berg's RUDP in order to set up an RUDP connection with **remote machine 30'**.
(*Id.*)

As discussed, *Wookey* describes using "code" to cause **client machine 10**, e.g., a web browser application 280, to use a protocol that can be separate from the one used to connect **client machine 10** and **remote machine 30**, in order to set up a second protocol connection between **client machine 10** and **remote machine 30'**. *Wookey* also describes the types of characteristics concerning the protocol that can be used for the second connection, such as negotiating connection parameters, reducing unintentional terminations, detecting/handling disconnections, specifying a permissible inactive time before connection termination. (EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153]; *supra* Section IX.A.1.f.) A POSITA would have been thus motivated by *Wookey*'s disclosures/suggestions to consider and incorporate a protocol that would provide the desired features to *Wookey* for the second protocol connection with **remote machine 30'**, such as the RUDP described by *Berg*. (EX1002 ¶180.)

Using the RUDP protocol described in *Berg* with *Wookey*'s system/method would have provided **client machine 10** with an improved reliable connection to **remote machine 30'** with the versatility of negotiated timeout parameters consistent with *Wookey*'s desired characteristics for the second connection. *See KSR*, 550 U.S.

Petition for *Inter Partes* Review
Patent No. 10,069,945

416–17. Indeed, by using a protocol like *Berg*’s RUDP protocol, *Wookey*’s method would enable negotiation of properties like the timeout values on a per-connection basis over a reliable connection (EX1007, 24:40–47), and provide alerts to the nodes if the session fails (*id.*, 2:55–60). *Wookey*’s method for “providing a client with a reliable connection to a host service” (EX1005 ¶[0077]) would therefore have been improved by incorporating *Berg*’s teachings. (EX1002 ¶181.) Accordingly, such a combination would have yielded the predictable result of providing a network communication exchange that minimizes disruptions caused by timeout issues. (*Id.* ¶184.) *See KSR*, 550 U.S. at 417.

It would have been a foreseeable and straightforward implementation to configure *Wookey*’s **client machine 10** such that the “code” (e.g., the encoded URLs in HTML page 288 or the HTML page 288 including such URLs), when used, would cause **client machine 10** to operate in accordance with a second protocol separate from the TCP like *Berg*’s RUDP. (EX1002 ¶182.) *See KSR*, 550 U.S. at 416. For example, as noted above, “*the URL includes a file, or a reference to a file, that contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (EX1005 ¶[0216].) In light of the above, a POSITA would have been motivated to configure *Wookey*’s encoded URL such that it includes information causing the client to establish a connection with **remote**

Petition for *Inter Partes* Review
Patent No. 10,069,945

machine 30' using a second protocol separate from the TCP (like *Berg's* RUDP). (EX1002 ¶183.) This implementation would have been achieved by combining known network design concepts and technologies described above by *Wookey* and *Berg*, and known in the art at the time, to obtain the foreseeable result of a reliable connection between the nodes in *Wookey*. (*Id.* ¶183.)

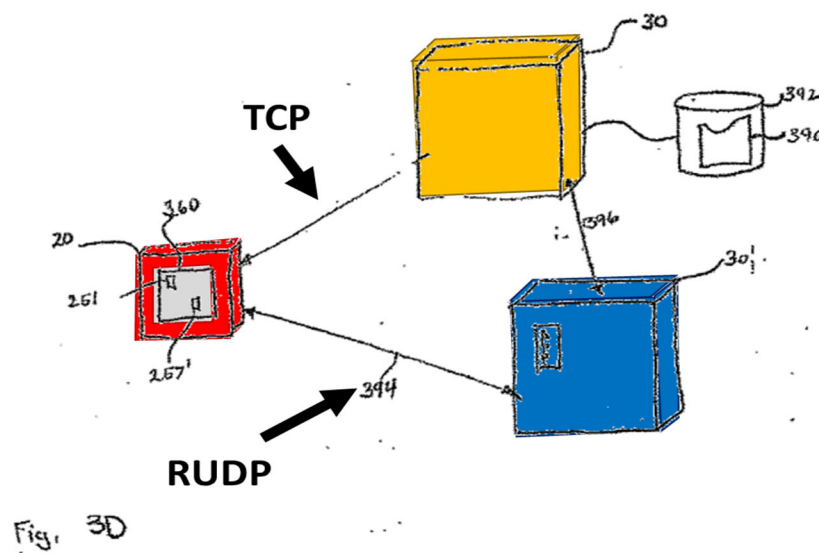
Indeed, the above-modification would have involved the substitution of features from one reliable protocol from a finite number of available alternative reliable communication protocols, such as that described by *Berg*. (*Id.* ¶185.) Thus, a POSITA would have had a reasonable expectation of success in the above-modification. (*Id.*) See *Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1364 (Fed. Cir. 2007) (“only a reasonable expectation of success, not a guarantee, is needed” in an obviousness analysis).¹²

A POSITA would have been skilled and knowledgeable to configure the above modification in various ways, while taking into account any known

¹² There is no evidence of nonobviousness at this time that would overcome the overwhelming evidence of obviousness as presented herein. Petitioner reserves the right to address any such evidence/argument should PO later present any in this proceeding.

Petition for *Inter Partes* Review
 Patent No. 10,069,945

programming, design, and other related concepts, limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. (EX1002 ¶186.) For example, a POSITA would have been motivated based on such disclosures to configure *Wookey's* system and process consistent with the non-limiting example reflected below.



(EX1005, FIG. 3D (annotated); EX1002 ¶186.)

* * *

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 1.f. (*Id.* ¶¶187–188; *see also infra* Sections IX.A.1.g–j.)

g) receiving a packet,

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶189–194.) As explained below, the “code” in the combined *Wookey-Berg* system/method would, when used by **client machine 10**, result in **client machine 10** using web browser application 280 to operate in accordance with the RUDP protocol to receive, by **client machine 10** from **remote machine 30’**, a responsive SYN segment (“packet”) during the setup of the RUDP connection. (*Id.* ¶190; *see also supra* Section IX.A.1.f.)

For example, “RUDP is designed to allow characteristics of each connection to be individually configured.” (EX1007, 17:42–46.) Thus, when a client initiates a connection, it first sends “a SYN segment which contains the negotiable parameters defined by the [Upper Layer Protocol] ULP via the [Application Programmer Interface] API.” (*Id.*, 24:40–42; *see also id.*, 18:56–62; EX1002 ¶191.) The server, upon receiving the client’s SYN segment, determines to either “accept” the proposed parameters in the received SYN segment or propose different parameters. (EX1007, 24:42–44.) In either case, the server returns the parameters in its responsive SYN segment (“packet”) to the client. (*Id.*) The client, upon receiving the server’s SYN segment (“receiving a packet”), can then choose to accept the parameters sent by the server by responding to the server with an ACK

Petition for *Inter Partes* Review
Patent No. 10,069,945

message to establish the connection, or it can send a reset (RST) segment to the server to refuse the connection. (*Id.*, 24:44–47; EX1002 ¶192.)

Berg explains that the messages sent to set up the RUDP connection, including the SYN segments, are UDP packets that include an RUDP header. (EX1007, 17:59–18:20, 19:1–40 (SYN segment).) Therefore, the SYN segment response sent from **remote machine 30'** and received by **client machine 10** in the *Wookey-Berg* combination is a packet (e.g., a UDP packet with an RUDP header). (EX1002 ¶¶193–195.)

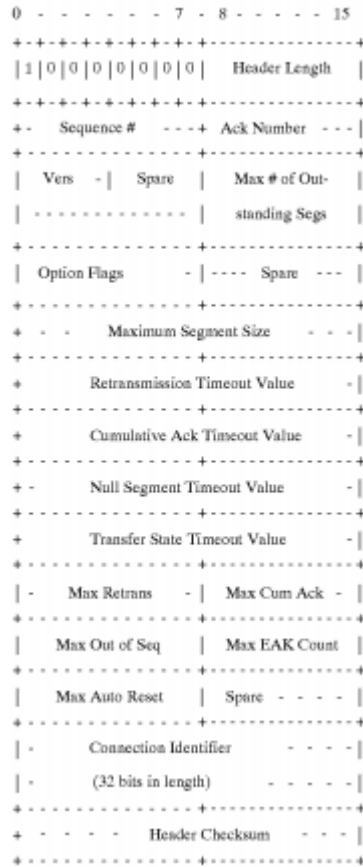
h) detecting an idle time period parameter field in the packet,

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶196–205.) For example, **client machine 10** in the combined *Wookey-Berg* system/method detects a null segment timeout parameter field (“an idle time period parameter field”) in the received SYN segment. (*Supra* Sections IX.A.1.f–g; EX1002 ¶197.)

For example, when the client receives the SYN segment sent from the server, it can *accept* or *reject* the proposed parameters (such as a null segment timeout value) in the received SYN segment. (EX1007, 24:44–48; EX1002 ¶198.) Figure

Petition for *Inter Partes* Review
Patent No. 10,069,945

2, below, of the RUDP specification,¹³ illustrates the SYN segment with the respective fields for its negotiable parameters for the RUDP connection:



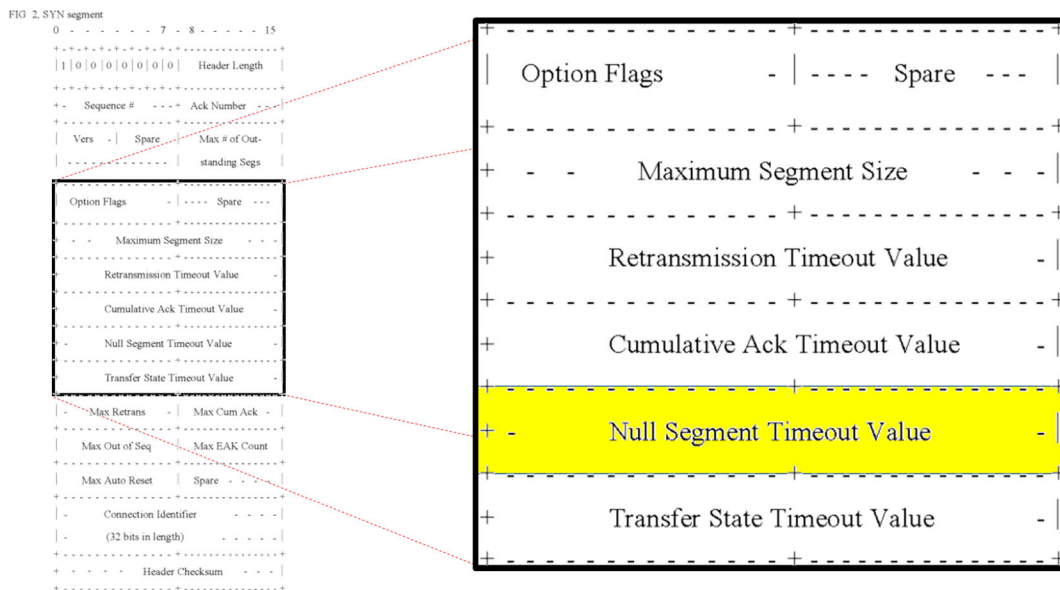
(*Id.*, 19:1–40; EX1002 ¶199; *see also* EX1002 ¶200.)

Berg further provides that the null segment timeout value is specified in milliseconds from a null segment timeout value (“idle time period”) parameter field

¹³ The RUDP specification is included in *Berg* as “Appendix 1.” (EX1007, 16:59–25:54; *see also* EX1011, 6.)

Petition for *Inter Partes* Review
 Patent No. 10,069,945

in the SYN segment. (EX1007, 18:58–60, 19:22–25, 20:36–42, 22:25–33.)



(*Id.*, 19:1–40 (excerpted and annotated); EX1002 ¶¶201; *see also* EX1002 ¶¶202–203.)

The *Wookey-Berg* combination would have thus been configured such that **client machine 10** detects the null segment timeout parameter field, in order to accept/reject the proposed null segment timeout value. (EX1002 ¶¶204–205; *see also supra* Section IX.A.1.f.ii, *infra* Section IX.A.1.i (discussing identification of the null segment timeout value in the SYN segment).)

Petition for *Inter Partes* Review
Patent No. 10,069,945

i) identifying metadata in the idle time period parameter field for an idle time period, where, after the idle time period is detected, the second protocol connection is deemed inactive,

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶206–220.) As discussed below, the *Wookey-Berg* combination discloses identifying a null segment timeout value (“metadata”) in the null segment timeout parameter field for a null segment timeout (“idle time period”), where, after the null segment timeout is detected, the RUDP connection is deemed inactive. (*Id.*, ¶¶207–208.)

For example, the client detects the null segment timeout parameter field in the SYN segment sent from the server so it can *accept* or *reject* the proposed null segment timeout value contained therein. (*See supra* Section IX.A.1.h.) Thus, in order to *accept* or *reject* the proposed null segment timeout value in the null segment timeout parameter field, the client necessarily identifies it. (EX1002 ¶¶209–210.)

To the extent the *Wookey-Berg* combination does not disclose the claimed “identifying,” it would have been rendered obvious in the combined *Wookey-Berg* system/process. (*Id.* ¶211.) A POSITA would have been motivated to configure the *Wookey-Berg* system/process to include functionality that identifies this parameter to facilitate the subsequent process of accepting or rejecting it in a manner consistent

Petition for *Inter Partes* Review
Patent No. 10,069,945

with the features described by *Berg*. (EX1007, 24:44–48; EX1002 ¶212; *see also* EX1002 ¶¶213–214.)

The '945 patent confirms the understanding that the null segment timeout value is metadata, explaining that metadata can be a value indicating a “duration of time” (EX1001, 21:60–65), which is what the above-discussed timeout value in *Berg* contains. (EX1007, 20:36–47; EX1002 ¶215.) Moreover, the above understanding is consistent with PO’s view of the same claimed features. (EX1017, 13 (PO alleging “identifying metadata (e.g., a value, etc.) in the idle time period parameter field (e.g., idle_timeout parameter field, etc.)”).)

The *Wookey-Berg* combination further discloses that the null segment timeout value is for an “idle time period, where, after the idle time period is detected, the second protocol connection is deemed inactive.” For instance, the null segment timeout value indicates the amount of time (e.g., milliseconds) to wait before sending a null segment when the connection has been idle (e.g., when a data segment has not been sent or received). (EX1007, 20:36–40.)¹⁴ The null segment is sent to determine “if the other side of a connection is still active.” (EX1007, 22:25–33,

¹⁴ *Berg* uses “null segment,” “NUL segment,” and “keep-alive” interchangeably. (EX1007, 20:36–42, 22:25–33; EX1002 ¶217.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

20:3–40.) This waiting period (when the connection is idle) before the client sends a null segment to the server to verify whether the connection is active represents an “idle time period.” (EX1002 ¶216.)

Furthermore, as a result of detecting a null segment timer’s expiration, which is set based on the null segment timeout value, the RUDP connection is “deemed inactive,” as claimed. (EX1002 ¶217.) For example, *Berg* explains that the null segment mechanism’s purpose is to determine whether the “connection is *still active*.” (EX1007, 22:25–32.) *Berg*’s null segment mechanism utilizes a null segment timer to control when to send a null segment. (*Id.*, 23:45–49.) For example, the client will send a null segment to the server when the client’s null segment timer expires. The duration of the client’s null segment timer is set to the null segment timeout (“idle time period”) specified by the null segment timeout value. (*Id.*, 23:45–49.) That is, the null segment timeout value defines a duration of time that the connection will remain inactive before the client sends a null segment to see if the connection still exists. (*Id.*, 20:36–41, 22:25–32; EX1002 ¶218.) Accordingly, the identified null segment timeout value is for a null segment timeout, where, in response to detecting expiration of the null segment timer (which was set to the null segment timeout), the RUDP connection is deemed inactive (and thereby causing

Petition for *Inter Partes* Review
Patent No. 10,069,945

the client to send a null segment to the server to see if the connection is still active). (EX1002 ¶218.)

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been further configured to include **client machine 10** identifying the null segment timeout value in the null segment timeout parameter field for a null segment timeout, where, in response to expiration of the null segment timeout being detected, the RUDP connection is deemed inactive, as claimed. (EX1002 ¶219; *see supra* Section IX.A.1.f.ii.) Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 1.i. (EX1002 ¶220.)

j) creating or modifying, by the client computer and based on the metadata, a timeout attribute associated with the second protocol connection.

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶221–225.) As explained above and further explained below, **client machine 10** in the *Wookey-Berg* combination would have been configured to create or modify, based on the null segment timeout value (“metadata”), a null segment timer (“timeout attribute”) associated with the RUDP connection. (EX1002 ¶222; *see supra* Section IX.A.1.i.)

For example, the client determines the timeout value for a null segment timer associated with the RUDP connection based on the null segment timeout value.

Petition for *Inter Partes* Review
Patent No. 10,069,945

(EX1007, 20:36–38, 23:46–59; EX1002 ¶223; *see supra* Section IX.A.1.i.) As discussed above, the client’s null segment timer is set (“creat[ed] or modif[ied]”) based on the null segment timeout value. (EX1007, 23:45–59; *see also id.*, 20:36–41; EX1002 ¶223; *supra* Section IX.A.1.i.)

Accordingly, in the *Wookey-Berg* combination, **client machine 10** creates or modifies, based on the null segment timeout value (“metadata”), a null segment timer (“timeout attribute”) associated with the RUDP (“second protocol”) connection. (EX1002 ¶224; *see supra* Section IX.A.1.f.ii.) Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 1.j. (EX1002 ¶225.)

2. Claim 7

a) **The computer-implemented method of claim 1 wherein the access is caused to be provided to the code, by permitting the code to be received from the another server computer.¹⁵**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶226–230.) As explained below, **remote machine 30** permits the code to be received by **client machine 10** from **remote machine 30’**. (EX1002 ¶227; *see also supra* Section IX.A.1.f.)

¹⁵ *See supra* n.7.

Petition for *Inter Partes* Review
Patent No. 10,069,945

For example, *Wookey* discloses that in its system/method, instead of the **remote machine 30** responding to **client machine 10**'s request to provide a list of accessible resources, **remote machine 30** can forward the received request to **remote machine 30'** for processing the request. (EX1005 ¶[178] (“*remote machine 30 functioning as a web server* receives communications from the client machine 10 . . . [where] *the web server forwards the communications to a remote machine 30'.*”).) In such a scenario, *Wookey* explains that instead of **remote machine 30** itself providing an HTML page with encoded URLs identifying the resources accessible to **client machine 10**, **remote machine 30'** could perform those functions. (EX1002 ¶228; EX1005 ¶¶[191] (describing that upon receiving the **client machine 10**'s request, **remote machine 30** may inform **client machine 10** that **remote machine 30'** can process the request as the Resource Neighborhood (RN) application 241 is available on **remote machine 30'**, and **remote machine 30'** provides the results to **client machine 10**), [190] (explaining that the RN application 241 is the means by which **client machine 10** interacts with remote machines in server farm 38 to obtain information regarding accessible resources, i.e., the HTML page 288 with encoded URLs specifying the available resources as explained for limitation 1.f).) By allowing **client machine 10** to receive the “code” from **remote machine 30'** (e.g., by forwarding the client's request to **remote machine 30'** for

processing), **remote machine 30** allows (e.g., “permits”) **client machine 10** to receive the “code” from **remote machine 30’**. (EX1002 ¶229.)

Thus, the *Wookey-Berg* combination discloses and/or suggests the features of claim 7. (*Id.* ¶230.)

3. Claim 8

a) **The computer-implemented method of claim 1 wherein the access is caused to be provided to the code, by permitting the code to be received from yet another server that is different from the server computer and the another server computer.**¹⁶

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶231–242.) As discussed below, it would have been obvious to configure the *Wookey-Berg* system/method such that **remote machine 30** permits the code to be received by a management server (“yet another server”) that is different from **remote machine 30** and **remote machine 30’** based on *Wookey*’s teachings and a POSITA’s knowledge. (EX1002 ¶232 *see also supra* Sections IX.A.1.f, IX.A.2.)

For example, *Wookey* discloses that one remote machine in server farm 38 may be designated as a “management server,” and that server will maintain

¹⁶ *See supra* n.7.

Petition for *Inter Partes* Review
Patent No. 10,069,945

information about the other remote machines (e.g., remote machines 30, 30', 30'', 30''' in Figure 1) in the server farm 38 and respond to client's requests for resources hosted by the remote machines. (EX1005 ¶[0137]¹⁷; *see also id.*, ¶¶[0231]–[0234] (similarly describing a centralized cache storing resource-related information of the remote machines), [0241]–[0244], [1204]–[1205], FIG. 1.) The management server (“yet another server”) can be any remote machine in server farm 38, i.e., a remote machine different from **remote machine 30** and **remote machine 30'**. (EX1005 ¶[0137]; EX1002 ¶233; *see also* EX1002 ¶¶234–235.) *See, e.g., Unwired Planet, LLC*, 841 F.3d at 1002; *Power Integrations, Inc.*, 843 F.3d at 1336-37.

Wookey discloses that upon receiving a client's request for accessing resources on remote machines in the server farm 38, **remote machine 30** can access a centralized database like the above-discussed management server to retrieve information regarding the most up-to-date resources available in the remote machines so it can provide the requested information to **client machine 10**. (EX1005 ¶¶[0141]–[0142] (describing a “*logical, common database (i.e., the*

¹⁷ *Wookey* also refers to the management server as a “management node,” “management process,” or “management component.” (EX1005 ¶¶[0137], [0144] [0181].)

dynamic store) that is *accessible by all of the remote machines 30* in the machine farm 38 *for accessing and storing some types of data*” referred to as “*runtime data*” such as “the number of virtual machines supported by a remote machine 30, the identity of the operating systems supported by a remote machine 30”), [0144]–[0147] (describing that the centralized dynamic store can operate like the management server and that “*all other remote machines 30 in the machine farm 38 communicate with the server acting as the centralized data store when seeking access to that runtime data*”), [0180]–[0181] (similar); EX1002 ¶236.)

Wookey, however, does not explicitly disclose a scenario where **remote machine 30**, upon receiving a client’s request for accessing resources, forwards the request to the management server so the management server responds to the client’s request by providing access to the HTML page 288 with encoded URLs to **client machine 10**. (EX1002 ¶237.) But it would be obvious to configure the *Wookey-Berg* system/method such that it would operate this way in light of *Wookey*’s teachings and a POSITA’s knowledge. (*Id.*)

For instance, as discussed above for claim 7, *Wookey* discloses that **remote machine 30** can forward the client’s request to another **remote machine 30’** so **remote machine 30’** can process the client’s request and return the requested information (i.e., the HTML page 288 with encoded URLs) to **client machine 10**.

Petition for *Inter Partes* Review
Patent No. 10,069,945

(*See supra* Section IX.A.2.) As such, *Wookey* already envisions a scenario where the client's request received by **remote machine 30** can be processed by another remote machine in farm 38. A POSITA would have been similarly motivated to configure the *Wookey-Berg* combination such that **remote machine 30** forwards the client's request to the management server so the management server can process the client's request and return the requested information to **client machine 10**, thereby allowing ("permit[ting]") **client machine 10** to receive the "code" from the management server ("yet another server") that is different from **remote machine 30** and **remote machine 30'**. (EX1002 ¶238.) Such an implementation would have had advantages and been a predictable and straightforward result achieved by combining well-known concepts disclosed by *Wookey* itself. (*Id.*) *See KSR*, 550 U.S. at 416–18.

A POSITA would have been so motivated because allowing the management server to directly respond to **client machine 10** with the information regarding accessible resources on remote machines in the server farm 38 would have further ensured that the most up-to-date information is being provided to the client. (EX1002 ¶239.) Indeed, a centralized server like the management server stores dynamic data that is "typically queried or *changed frequently* during runtime." (EX1005 ¶[0142].) Thus, if the management server directly responds to **client**

Petition for *Inter Partes* Review
Patent No. 10,069,945

machine 10, it can provide the most up-to-date information (i.e., the HTML page with encoded URLs identifying the accessible resources). (*Id.*; EX1002 ¶239.) Accordingly, such a modification to the *Wookey-Berg* combination would have yielded the predictable and desirable result of providing **client machine 10** with the latest information regarding resources it can access on server farm 38. (EX1002 ¶240.); *see KSR*, 550 U.S. at 417. The above-modification would have involved choosing among a finite number of servers that could have responded to the client's request, e.g., **remote machine 30**, **remote machine 30'**, or yet another remote machine in server farm 38 that would have been designated as a management server as discussed above. (EX1002 ¶240.)

A POSITA would have been skilled and knowledgeable to configure the combined *Wookey-Berg*'s method such that a management server that is different than **remote machine 30** and **remote machine 30'** responds to **client machine 10**'s request by providing it the HTML page with encoded URLs, while considering any known programming, design, and other related concepts, limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. (EX1002 ¶241.) Indeed, *Wookey* itself discloses that the management server can “respond to requests for access to resources hosted by remote machines 30.” (EX1005 ¶[0137].) Thus, a POSITA would have had a reasonable expectation of

Petition for *Inter Partes* Review
Patent No. 10,069,945

success in the above-modification to the *Wookey-Berg* combination. (EX1002 ¶241.) *Pfizer*, 480 F.3d at 1364.

Thus, the *Wookey-Berg* combination discloses and/or suggests the features of claim 8. (EX1002 ¶242.)

B. Ground 2: The Combination of *Wookey*, *Berg*, and *Eggert* Renders Obvious Claims 11–12**1. Claim 11**

a) The computer-implemented method of claim 1 wherein the code causes the client computer to operate such that the timeout attribute is subject to a global setting.

Wookey in combination with *Berg* and *Eggert* discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶243–262.) As discussed above, in the *Wookey-Berg* combination, *Wookey*’s **client machine 10** creates or modifies the null segment timer (“timeout attribute”) associated with the RUDP connection based on the negotiated null segment timeout value (“metadata”). As discussed below, it would have been obvious to configure the “code” to cause **client machine 10** to operate such that the null segment timer is subject to a default null segment timeout value (i.e., a “global setting”), e.g., when **client machine 10** and **remote machine 30’** cannot agree upon the null segment timeout value to use for setting the null segment timer. (EX1002 ¶245.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

For example, as explained above, *Berg* discloses the server may accept the client's proposed null segment timeout value or it may propose a different null segment timeout value. (*See supra* Sections IX.A.1.i–j; EX1007, 24:39–44.) Where the server proposes a different null segment timeout value, the client can choose to accept this value or refuse the connection. (EX1007, 24:44–47.) *Berg*, however, is silent regarding allowing the RUDP connection to be established where the client and server do not agree upon the null segment timeout value. (EX1002 ¶246.)

Based on the teachings of *Eggert* and a POSITA's knowledge, however, it would have been obvious to configure the code such that it would cause **client machine 10** in the *Wookey-Berg* combination to operate such that the null segment timer is subject to a global setting, e.g., when the client and remote machines 10, 30' do not agree on the null segment timeout value. (*Id.* ¶247.)

i. Eggert

Eggert, like *Wookey* and *Berg*, relates to establishing a reliable connection between two nodes. (EX1006, 1–3; EX1002 ¶¶98–103; *infra* Section IX.B.1.a.ii (discussing similarities between *Wookey*, *Berg*, and *Eggert*).) *Eggert*, like *Berg*, discloses a technique that allows nodes to share and negotiate connection-specific parameters (e.g., timeout values) to be used when establishing a connection between two nodes. (EX1006, Abstract; EX1007, 17:43–47; *supra* Section IX.A.1.f.i

Petition for *Inter Partes* Review
Patent No. 10,069,945

(discussing *Berg*.) For instance, *Eggert* discloses a communication protocol that allows nodes to negotiate a connection-specific abort timeout. (EX1006, 3; EX1002 ¶248.)

Eggert discloses that during the setup of a connection, when the nodes negotiating the value of an abort timeout cannot agree on the abort timeout value, the abort timeout for the connection is subject to a default value. (EX1006, 5–7.) For example, *Eggert* discloses that when the abort timeout value proposed by a node (e.g., a client) initiating the negotiation is not acceptable to the other node (e.g., a server), the other node may opt for the nodes to use a default timeout value for the connection. (*Id.*, 7.) The responding node does this by rejecting the offered timeout value from the initiating node (e.g., client). (*Id.*) Thus, when the responding node rejects the initiating node’s proposed abort timeout value, i.e., when the nodes cannot agree upon an abort timeout value during the negotiation, the abort timeout for the connection is subject to a default abort timeout. (*Id.*) (EX1002 ¶249.) Using the default abort timeout value for the connection when the server rejects the client’s proposed timeout value would have ensured that nodes still establish a connection with no additional negotiation processes (e.g., without renegotiating the connection). (*Id.* ¶250.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

Eggert's process thus discloses the features of subjecting a timeout attribute of a connection to a default setting, e.g., when the nodes negotiating the value of an abort timeout cannot agree on the abort timeout value. (EX1002 ¶251.)

ii. Reasons to Combine

A POSITA would have been motivated in light of *Eggert* to configure the code in the combined *Wookey-Berg* method causing **client machine 10** to operate such that its null segment timer ("timeout attribute") is subject to a default null segment timeout value ("global setting"), e.g., when **client machine 10** and **remote machine 30'** cannot agree upon the null segment timeout value for setting the null segment timer. (EX1002 ¶252.) Such an implementation would have been a predictable and straightforward combination of well-known technologies using known methods. *See KSR*, 550 U.S. at 416–18.

For example, a POSITA would have been motivated to implement *Eggert*'s teachings regarding using a default timeout value ("global setting"), e.g., when **client machine 10** and **remote machine 30'** cannot agree upon the null segment timeout value, because it would have allowed for the combined process/system to establish a connection to accommodate **client machine 10** to access the requested resource based on a default null segment timeout value for the connection, without requiring the machines to renegotiate the connection parameters. (EX1002 ¶253;

Petition for *Inter Partes* Review
Patent No. 10,069,945

EX1005 ¶¶[0154], [0174]; EX1007, 24:39-47.) The default null segment timeout value is a global setting because, e.g., it would be used as the default setting for any RUDP connection negotiation process where the nodes do not reach an agreement regarding the value of the duration of the null segment timeout specific to the connection being negotiated, i.e. the default setting would have been globally applicable to multiple connections. (EX1002 ¶254.) Alternatively, the default null segment timeout value is a global setting because it would have been “global” at least for the pair of nodes (**client machine 10** and **remote machine 30**) negotiating the null segment timeout value. (*Id.*)

A POSITA facing the wide range of needs created by developments in the technical field of *Wookey*, *Berg*, and *Eggert*, would have appreciated the benefits of allowing the machines to settle to a default null segment timeout value to establish a connection in the *Wookey-Berg* combination where they cannot reach an agreement regarding the value. (EX1002 ¶255.) Indeed, the use of global settings for connection timeout parameters (such as a global setting for the null segment timeout value in the combined *Wookey-Berg* process/system) was a well-known concept in the networking field before the '945 patent. (EX1002 ¶255; EX1008, 9 (explaining that a protocol may include “default values to be used when” certain “features are not needed”), 49 (explaining that a user timeout has a “global default”

Petition for *Inter Partes* Review
Patent No. 10,069,945

of “five minutes”).) Moreover, when implementing processes involving negotiation of connection-related parameters during connection establishment in a UDP-based implementation such as *Berg*, a POSITA would have been motivated to consider relevant teachings from TCP-based implementations such as *Eggert* given that TCP and UDP were the most widely used communication protocols in the networking field at the time of the ’945 patent. (EX1002 ¶256; EX1006, 1–7, EX1007, 8:8–18, 16:59–17:17; *see also supra* Section IX.A.1.f (discussing *Wookey*’s disclosures regarding both TCP-based implementations and implementations based on protocols other than TCP).)

Therefore, based on *Eggert*’s disclosure and knowledge in the art, a POSITA would have been motivated, and would have found it a commonsense approach, to configure the *Wookey-Berg* combination such that the null segment timeout value would have been subject to a global default timeout value when **remote machine 30’** and **client machine 10** do not agree upon the null segment timeout value, in order to allow the two machines to establish a connection. (EX1002 ¶257.) Such a modification would have been compatible with RUDP, which “is designed to allow characteristics of each connection to be individually configured so that many protocols with different transport requirements can be implemented simultaneously on the same platform.” (EX1007, 17:43-47.) For example, when **client machine**

Petition for *Inter Partes* Review
Patent No. 10,069,945

10 and **remote machine 30'** agree upon a null segment timeout value during the negotiation in the RUDP connection establishment process, the agreed-upon value would override any default timeout value. (*Id.*; EX1002 ¶257.) And when **client machine 10** and **remote machine 30'** cannot agree upon a null segment timeout value during the negotiation, a default null segment timeout value would have been used the same way a negotiated (agreed-upon) null segment timeout value would have been used to set the null segment timers on the client side and server side. (EX1007, 23:45–52.) That is, the client's null segment timer would have been set to the default null segment timeout value, and the server's null segment timer would have been set to two times the default null segment timeout value. (*Id.*; EX1002 ¶257.)

Thus, the proposed modification of the *Wookey-Berg* system/method in light of *Eggert's* teachings would have been consistent with *Berg's* goal of allowing the RUDP connection parameters to be modified on a per-connection basis (e.g., when the nodes agree on the parameters for the connection), while providing the added benefit of allowing an RUDP connection to still be established in the case when the nodes do not agree upon a null segment timeout value so that **client machine 10** can still access its desired resource(s). (EX1007, 17:40–47; EX1005 ¶¶[0154], [0174]; EX1002 ¶258.) And as discussed above, *Wookey*, *Berg*, and *Eggert* all relate to

Petition for *Inter Partes* Review
Patent No. 10,069,945

establishing reliable connections between nodes, and both *Wookey* and *Berg* disclose techniques allowing for the nodes to negotiate connection-specific parameters. Accordingly, a POSITA would not be detracted from further modifying the *Wookey-Berg* system/method based on *Eggert*'s teachings, as proposed. (EX1002 ¶258; *see also id.* ¶259.)

The combination of *Wookey* and *Berg* with *Eggert*, similar to the *Wookey-Berg* combination, would have involved the use of known technologies (e.g., aspects of similar protocols including the negotiation of timeout values) and design concepts and processes to obtain the foreseeable result of using a default null segment timeout value when **client machine 10** and **remote machine 30'** do not reach an agreement for a null segment timeout value during the negotiation, when establishing the RUDP connection. (EX1002 ¶260.) Indeed, the RUDP specification provides that the "recommended value of the null segment timer is 2 seconds." (EX1011, 13–14; EX1002 ¶260.) Thus, a POSITA could have considered such teachings when further configuring the *Wookey-Berg* combination in light of *Eggert*'s teachings to subject the null segment timer to a default value as it would have been a practical, common sense approach to ensure that **client machine 10** establishes a connection to **remote machine 30'** in order to access the requested resource(s). (EX1002 ¶260.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

As such, a POSITA would have been skilled and knowledgeable to configure the *Wookey-Berg*'s system/method to implement a default setting for a null segment value should the **client machine 10** and **remote machine 30** not agree upon a null segment timeout value, while taking into account any known programming, design, and other related concepts, limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. (*Id.* ¶261.) Thus, a POSITA would have had a reasonable expectation of success in the above-modification. (*Id.*) See *Pfizer, Inc.*, 480 F.3d at 1364.

Accordingly, the *Wookey-Berg-Eggert* discloses and/or suggests the limitations of claim 11. (EX1002 ¶262.)

2. Claim 12

a) **The computer-implemented method of claim 1 wherein the code causes the client computer to operate such that the timeout attribute is subject to a connection-specific setting that is capable of overriding a global setting.**

The *Wookey-Berg-Eggert* combination discloses and/or suggests and renders obvious this limitation for at least the same reasons discussed above for claim 11. (EX1002 ¶¶263–264.) As explained above, the “code” in the *Wookey-Berg-Eggert* combination causes **client machine 10** to operate such that the null segment timer is subject to a default null segment timeout value when **client machine 10** and **remote machine 30** do not agree upon a null segment timeout value during the negotiation

Petition for *Inter Partes* Review
Patent No. 10,069,945

of connection parameters in the RUDP establishment process. (*See supra* Section IX.B.1.) As also explained above, when **client machine 10** and **remote machine 30'** reach an agreement for a null segment timeout value during the negotiation (when establishing the RUDP connection), then the null segment timer (“timeout attribute”) is subject to the agreed-upon timeout value for the RUDP connection (“a connection-specific setting”), which would thereby override the default null segment timeout value (e.g., “a global setting”). (*See supra* Section IX.B.1.) Thus, the negotiated, agreed-upon null segment timeout value during the RUDP connection establishment process is capable of overriding the default null segment timeout value in the *Wookey-Berg-Eggert* combination. (*Id.*; EX1002 ¶264.)

X. EGGERT IS A PRINTED PUBLICATION

Eggert is an IETF Internet-Draft (“ID”) and is prior art under at least 35 U.S.C. §102(b) (pre-AIA) because it was published in April, 2004. The declaration of Alexa Morris, Managing Director of IETF, confirms *Eggert* was published, disseminated, and reasonably available to the public by April, 2004. (EX1031 ¶¶1-3, 9–10.)

IDs were at the time, and continue to be, working documents published through the IETF Secretariat and disseminated to the public by IETF through various media so that others may comment on them. (*Id.* ¶¶4–6.) Since 1998 (including

Petition for *Inter Partes* Review
Patent No. 10,069,945

2004 and today), the IETF Secretariat publishes an ID on its public website, and publication announcements were sent to an IETF mailing list and relevant working group mailing lists. (*Id.* ¶¶5–6.) Anyone could have subscribed to IETF mailing lists, and the archives of all IETF mailing lists are publicly available on IETF’s website. (*Id.* ¶¶6–8.)

Eggert was published by the IETF in April, 2004. (*Id.* ¶¶9–10.) In 2004, a POSITA could have learned about *Eggert* in various ways, such as through announcements on the IETF announce mailing list, discussions on the IETF tcpm working group mailing list, or review of the archives of the IETF announce or tcpm mailing lists. (*Id.*)

Collectively, this demonstrates that *Eggert* was published and publicly available in 2004 and at least prior to 2010. *Polycom, Inc., v. Directpacket Research, Inc.*, IPR2019-01235, Paper 19, 30–32 (Jan. 13, 2020) (finding a similar IETF ID was publicly available based on similar evidence).

XI. DISCRETIONARY DENIAL IS NOT APPROPRIATE

A. *Eggert* is Not Cumulative

Eggert is not cumulative to any prior art the Examiner considered during prosecution of the ’945 patent and no arguments similar to those contained herein were ever presented to or considered by the Office. Nonetheless, PO may assert here

Petition for *Inter Partes* Review
Patent No. 10,069,945

(as it did in the Google -845 IPR) that RFC 5482 (EX1034) listed on the face of the '945 patent is substantially similar to *Eggert*. It is not, as the Board found in the Google -845 IPR. Google -845 IPR, Paper 16 at 14–17. The Board should reach the same conclusion here.

First, the Board routinely institutes trial where references in an IPR were considered in an IDS but not relied upon to reject claims during prosecution. *See, e.g., Amber.IO, Inc. D/B/A Two Tap v. 72Lux, Inc. D/B/A Shoppable*, IPR2020-00015, Paper 8 at 18-20 (April 1, 2020) (declining to exercise discretion where references were cited in an IDS but no evidence that the references were substantively considered) (citing *Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6 (February 13, 2020) (precedential)); *Apple, Inc. v. Omni Medsci, Inc.*, IPR2020-00029, Paper 7 at 52–55 (April 22, 2020) (similar). Consistent with those proceedings, RFC 5482 was cited during prosecution along with twenty other references in an IDS with no explanation regarding any relevant disclosures. (EX1004, 101–104.) Moreover, RFC 5482 was never relied upon by the Examiner to reject the claims. (*Id.*, 56–60.) Even assuming *Eggert* is cumulative to RFC 5482, which it is not as explained below, *Eggert* was not considered in the light being presented herein (e.g., with supporting expert testimony).

Petition for *Inter Partes* Review
Patent No. 10,069,945

Second, as the Board previously found, *Eggert*'s protocol is fundamentally different from that discussed in RFC 5482 because, in contrast with RFC 5482, *Eggert*'s protocol calls for a common negotiated value for the timeout. Google -845 IPR, Paper 16 at 14–17. While *Eggert* discloses a negotiation protocol, RFC 5482 was purposefully designed such that “an exchange of UTO [user timeout] options between both ends of a connection is not a binding negotiation.” (EX1034, 4.)

Eggert also has meaningful disclosures not included in RFC 5482. For example, it is *Eggert*'s mechanisms disclosed with reference to Figure 2 regarding exchange of abort timeout information that is relied upon to teach the claim features relating to a timeout attribute being subject to a global setting if the nodes cannot agree during negotiation. (*Supra* Sections IX.B.1–2.) In contrast, RFC 5482 “does not define a mechanism to negotiate support of the TCP User Timeout Option [UTO] during the three-way handshake.” (EX1034, 9.) Therefore, these “material differences between the asserted art and the prior art involved during examination” weigh against discretionary denial. *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8, 17-18 (Dec. 15, 2017) (precedential as to §III.C.5, first paragraph); *see also* Google -845 IPR, Paper 16 at 16–17. Thus, *Eggert* is not cumulative to RFC 5482.

Accordingly, Petitioner requests that the Board institute review.

Petition for *Inter Partes* Review
Patent No. 10,069,945

B. The Related Litigation Provides No Basis For Discretionary Denial

The Board should not exercise its discretion to deny institution under 35 U.S.C. § 314(a). *NHK Spring Co., Ltd. v. Intri-Plex Techs, Inc.*, IPR2018-00752, Paper 8 (Sept. 12, 2018) does not apply here as no trial date has been scheduled in the related litigation. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 3 (Mar. 20, 2020) (precedential). The six-factor test addressed in *Fintiv* (“*Fintiv* factor(s)”) favors institution. *See id.*, 5–6.

The **first *Fintiv* factor** (stay) is neutral, if not weighing in favor of institution. While there is currently no stay pending IPR resolution before the Western District of Texas (“WDTX”) court, Google intends to pursue such a stay should institution be granted. The Board should not speculate as to how the court would rule on the requested stay based on actions taken in different cases. *Western Digital Corp. et al. v. Martin Kuster*, IPR2020-01391, Paper 10 at 8–9 (February 16, 2020) (finding this factor neutral despite Patent Owner assertions regarding WDTX Judge Albright’s past practices regarding stays). Moreover, Google filed a motion to transfer in the district court litigation. The Court has currently stayed the litigation pending resolution of Google’s transfer motion. (EX1030.)

The **second *Fintiv* factor** (proximity of trial dates) weighs strongly in favor of institution. *First*, the district court has not set a trial date, which “weighs

Petition for *Inter Partes* Review
Patent No. 10,069,945

significantly against exercising [] discretion to deny institution of the proceeding.” *Google LLC v. Uniloc 2017 LLC*, IPR2020-00441, Paper 13 at 35 (July 17, 2020); *Nested Bean, Inc. v. Big Beings USA PTY LTD*, IPR2020-01234, Paper 15 at 14 (Jan. 25, 2021) (same). In fact, while the parties submitted proposed schedules, the court granted Google’s motion to stay the case pending resolution of Google’s transfer motion and declined to set a case schedule.¹⁸ (EX1030; EX1024.) Accordingly, determining at this time whether the Board’s anticipated FWD date (e.g., October–November 2022) will occur after trial in the related litigation would be speculative.

Even if the parties’ most recent proposed schedule was considered as a baseline for comparison (which Google submits would be improper at this time), dates in that proposed schedule would need to be delayed because they were proposed prior to the court’s recent stay order and did not take the stay into consideration. (EX1030; EX1024, 5–9.) Accordingly, the proposed dates in light

¹⁸ Disposition of Google’s transfer motion has taken priority over other activities. (EX1030.) *See also In re Apple Inc.*, 979 F.3d 1332, 1337 (Fed. Cir. 2020) (“disposing of a filed transfer motion ‘should unquestionably take top priority.’”).

Petition for *Inter Partes* Review
Patent No. 10,069,945

of the court’s stay order demonstrate that trial will likely occur after August 2022.¹⁹ (EX1024, 6, 9.) Thus, any projected trial date (*albeit* speculative) would be more likely after the expected due date of the Board’s FWD (e.g., around October–November 2022).

The Board has instituted trial even where an *actual* litigation trial date precedes the FWD deadline by two months or more. *Sand Revolution II, LLC v. Cont’l Intermodal Grp–Trucking LLC*, IPR2019-01393, Paper 24 at 8–10 (June 16, 2020) (“*Sand*”) (informative) (factor 2 favored institution despite WDTX trial preceding FWD deadline by five months); *Western Digital Corp.*, IPR2020-01391 at 9-10; *Apple Inc. v. Parus Holdings, Inc.*, IPR2020-00686, Paper 9 at 11–13, 22; *Fintiv*, IPR2020-00019, Paper 11 at 5, 9.

It is also unlikely that trial in WDTX will proceed without delay. WDTX has issued a suspension order every month for the past twelve months suspending almost

¹⁹ Consistent with the court’s practice, the proposed order stages deadlines and trial date based on the proposed August 2021 *Markman* hearing and that the “actual trial date” may materially differ from the schedule and that the court would consider “reasonable amendments to the case schedule post-*Markman*.” (EX1024, 9 n.3; *id.*, 6.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

all trials in the district from March 13, 2020 to at least April 30, 2021 due to the COVID-19 pandemic, creating a large backlog of trials. (EX1025; *see also* EX1026, 2 (agreeing to stipulation to postpone trial date “to allow the COVID-19 situation to ameliorate”); EX1029 (showing J. Albright’s docket including 792 cases filed in 2020).) Thus, as the Board has recognized, civil trials in WDTX “may possibly slip ... [due to] months of backlogged trials, including many active criminal cases that would take precedence over civil trials.” *HP Inc. v. Slingshot Printing LLC*, IPR2020-01085, Paper 12 at 7 (Jan. 14, 2021); *Sand* at 8-10.

Finally, the potential for transfer based on Google’s transfer motion further reduces the likelihood that trial will precede the Board’s FWD. *Dish Network, L.L.C. v. Broadband iTV, Inc.*, IPR2020-01267, Paper 15 at 17–18 (Jan. 21, 2021) (explaining that the second *Fintiv* factor “is a proxy for the *likelihood* that the trial court will reach a decision on validity issues before the Board reaches a [FWD]”). Indeed, transfer of the WDTX case to the proper venue will undoubtedly delay any trial beyond the anticipated FWD due date. *C.f. Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 8, at 7 n.7 (Mar. 27, 2020) (denying institution based in part on scheduled trial date that was later vacated upon transfer); *see Uniloc 2017 LLC v. Google LLC*, Case No. 18-CV-00502, Dkt No. 277 (E.D. Tex. June 19, 2020) (transfer order vacating unreached deadlines).

Petition for *Inter Partes* Review
Patent No. 10,069,945

The **third *Fintiv* factor** (investment in parallel proceedings) also weighs in favor of institution. The case is in its earliest stages, and as discussed, all deadlines other than those concerning Google’s transfer motion have been stayed. Thus, investment by the parties in invalidity and claim constructions issues will be delayed, especially if the case is transferred. And even if the transfer motion is denied, using the above-discussed proposed schedule as an exemplary baseline (which will require adjustments), any *Markman* hearing in the case is likely not to occur until August 2021 or later, and thus would occur near the expected due date of the Board’s institution decision. (EX1024, 6.) Discovery will not commence, and dispositive motions will not be due, until after the *Markman* hearing. Other case activity that has taken place or will eventually take place (outside claim construction) that does not relate to validity should “not weigh in [the Board’s] consideration of this issue.” *Western Digital Corp.*, IPR2020-01391 at 11. Such minimal investment is insufficient to support discretionary denial. *Juniper Networks, Inc. v. Huawei Digital Techs. (Cheng Du) Co., Ltd.*, IPR2020-01130, Paper 13 at 12–13 (Jan. 22, 2021); *Dish* at 19–21; *HP* at 7. Additionally, Google’s diligence in filing this

Petition for *Inter Partes* Review
Patent No. 10,069,945

Petition just four months after receiving PO's narrowed list of asserted claims²⁰ further weighs against discretionary denial. *Dish* at 20–21; *Fintiv* at 11.

The **fourth *Fintiv* factor** (overlap) is neutral if not in favor of institution. There is, at this early stage of the district court litigation, no evidence of overlap with the grounds here and any invalidity positions to be pursued in the district court. Moreover, preliminary invalidity contentions have not yet been served. And, as discussed above, the district court must first resolve the pending transfer motion before addressing substantive issues, including invalidity. Given the stay of the litigation, it is likely expert discovery relating to invalidity will not be completed until at the earliest mid-2022 and dispositive motions on those issues would be due after that. Accordingly, substantive invalidity issues would not be addressed in the

²⁰ PO initially asserted 455 claims across eight patents (including 141 claims from the '945 patent). (EX1035, 1.) The WDTX court ordered PO to substantially reduce the asserted claims to approximately 65 claims. On October 20, 2020, PO narrowed its asserted claims as ordered, but “reserve[d] the right to either narrow the claims further, **substitute claims**, or **add a reasonable number of claims**.” (EX1027.) On December 4, 2020, PO informed Petitioner that it will not modify its narrowed list of asserted claims “**at this time**.” (EX1028.)

Petition for *Inter Partes* Review
Patent No. 10,069,945

litigation until well past the institution decision and likely very near the time of the Board's expected FWD. Moreover, Petitioner may rely on invalidity grounds not asserted here in the district court litigation, including system art and prior art references and combinations not raised herein. Therefore, any potential overlap between this proceeding and the district court is minimal and speculative at this time.

Other circumstances (***Fintiv* factor six**) also favor institution. The Petition presents strong grounds demonstrating the unpatentability of the challenged claims, which weighs against discretionary denial. (*See supra* Section IX.) *Western Digital Corp.*, IPR2020-01391 at 14-15. Indeed, the Board has already instituted the Unified -742 IPR (based on *Berg*) and the Google -845 IPR (based on *Eggert*) challenging claims of the related '995 patent. (*See supra* Section II.) Both *Berg* and *Eggert* are being applied here. Moreover, this petition and the concurrently-filed sister petition (*id.*) are the only challenges to the '945 patent before the Board, which is a "crucial fact" favoring institution. *Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 10 at 6 (May 12, 2020).²¹

²¹ Discretionary denial under *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper No. 19 is not applicable here given the

Petition for *Inter Partes* Review
Patent No. 10,069,945

While ***Fintiv* factor 5** (same parties) may weigh slightly in favor of denial, the remaining factors are at least neutral, if not strongly favoring institution. Further, even if the Board finds that **factor 1** (or even **factor 2**) somehow tilts in favor of denial, Petitioner’s diligence, the lack of relevant investment of resources, lack of evidence of overlap, and/or strength of Petitioner’s grounds (**factors 3, 4, 6**) outweigh these other factors. *See e.g., SK Hynix Inc. et al. v. Netlist, Inc.*, IPR2020-01421, Paper 10 at 6–13 (Mar. 16, 2021). Accordingly, based on a “holistic view” of whether integrity of the system and efficiency is best served, institution here is proper. *Samsung Elecs. Co. Ltd. v. Dynamics Inc.*, IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

’945 patent is not at issue in any other proceeding pending before the Board (outside of Petitioner’s concurrently-filed sister petition).

Petition for *Inter Partes* Review
Patent No. 10,069,945

XII. CONCLUSION

For the reasons given above, Petitioner requests institution of IPR for the challenged claims based on each of the specified grounds.

Respectfully submitted,

Dated: April 29, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,069,945

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,069,945 contains, as measured by the word-processing system used to prepare this paper, 13,983 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: April 29, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,069,945

CERTIFICATE OF SERVICE

I hereby certify that on April 29, 2021, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,069,945 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on PAIR:

Patrick Caldwell (Reg. No. 44580)
G. Gordon (Reg. No. 64517)
P.O. Box 59655
Dept. SVIPGP
Dallas, TX 75229

A courtesy copy was also sent via electronic mail to the Patent Owner's litigation counsel at the following addresses:

TD-PTAB@devlinlawfirm.com
ddahlgren@devlinlawfirm.com
dlflitparas@devlinlawfirm.com
1526 Gilpin Ave.
Wilmington, DE 19806

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)

EXHIBIT 9

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

JENAM TECH, LLC,
Patent Owner

Patent No. 10,306,026

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,306,026**

Petition for *Inter Partes* Review
Patent No. 10,306,026

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	1
III.	PAYMENT OF FEES	3
IV.	GROUND FOR STANDING.....	3
V.	PRECISE RELIEF REQUESTED	3
A.	Claims for Which Review Is Requested	3
B.	Statutory Grounds of Challenge.....	4
VI.	LEVEL OF ORDINARY SKILL	6
VII.	OVERVIEW OF THE '026 PATENT.....	6
VIII.	CLAIM CONSTRUCTION	10
IX.	DETAILED EXPLANATION OF GROUNDS.....	13
A.	Ground 1: <i>Eggert</i> Anticipates Claims 1 and 33	13
1.	Claim 1	13
2.	Claim 33	28
B.	Ground 2: The Combination of <i>Eggert</i> and <i>Abdolbaghian</i> Renders Obvious Claims 1, 17, 21–22, 33, 43, 45–48.....	44
1.	Claims 1 and 33.....	44
2.	Claim 17.....	46
3.	Claim 21	55
4.	Claim 22	55
5.	Claim 43	57
6.	Claim 45	57

Petition for *Inter Partes* Review
Patent No. 10,306,026

7.	Claim 46	59
8.	Claim 47	59
9.	Claim 48	61
C.	Ground 3: The Combination of <i>Eggert</i> and <i>Tucker</i> Render Obvious Claims 31–32 and 51	62
1.	Claim 31	62
2.	Claim 32	69
3.	Claim 51	71
D.	Ground 4: The Combination of <i>Eggert</i> , <i>Abdolbaghian</i> and <i>Tucker</i> Render Obvious Claims 31–32 and 51	71
1.	Claims 31–32, and 51	71
X.	EGGERT IS A PRINTED PUBLICATION	71
XI.	DISCRETIONARY DENIAL IS NOT APPROPRIATE	73
A.	<i>Eggert</i> is Not Cumulative.....	73
B.	The Related Litigation Provides No Basis For Discretionary Denial	75
XII.	CONCLUSION.....	83

Petition for *Inter Partes* Review
Patent No. 10,306,026

LIST OF EXHIBITS

EX1001	U.S. Patent No. 10,306,026
EX1002	Declaration of Bill Lin, Ph.D.
EX1003	Curriculum Vitae of Bill Lin, Ph.D.
EX1004	Prosecution History of U.S. Patent No. 10,306,026
EX1005	RESERVED
EX1006	L. Eggert, TCP Abort Timeout Option, draft-eggert-tcm-tcp-abort-timeout-option-00, Network Working Group, Internet-Draft (April 14, 2004) (“ <i>Eggert</i> ”)
EX1007	U.S. Patent No. 6,981,048 to Abdolbaghian (“ <i>Abdolbaghian</i> ”)
EX1008	DARPA RFC 793 TRANSMISSION CONTROL PROTOCOL
EX1009	RFC 1122 “Requirements for Internet Hosts -- Communication Layers”
EX1010	U.S. Patent No. 9,923,995
EX1011	RFC 1001
EX1012	U.S. Patent No. 6,212,175 to Harsch
EX1013	U.S. Patent No. 6,665,727 to Hayden (“ <i>Hayden</i> ”)
EX1014	U.S. Patent No. 7,636,805 to Rosenberg
EX1015	RFC 5440
EX1016	IETF RFC 5482
EX1017	<i>Jenam Tech, LLC’s Second Amended Set of Infringement Contentions</i> regarding U.S. Patent No. 10,306,026 (March 17, 2021)

Petition for *Inter Partes* Review
Patent No. 10,306,026

EX10181	<i>Jenam Tech, LLC v. LG Electronics, Inc. et al.</i> , Case No. 4:19-cv-00249, ECF No. 1 (E.D. Tex. Apr. 3, 2019)
EX1019	Declaration of Alexa Morris for Eggert
EX1020	U.S. Pre-Grant Publication No. 2004/0093376 to De Boor <i>et al.</i>
EX1021	U.S. Patent 7,535,913 to Minami <i>et al.</i>
EX1022	U.S. Pre-Grant Publication No. 2005/0204013 to Raghunath <i>et al.</i>
EX1023	U.S. Pre-Grant Publication No. 2007/0005804 to Rideout
EX1024	U.S. Pre-Grant Publication No. 2004/0098748 to Bo <i>et al.</i>
EX1025	IETF RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1
EX1026	U.S. Patent No. 8,259,716 to Diab
EX1027	Bova et al., RELIABLE UDP PROTOCOL <draft-ietf-sigtran-reliable-udp-00.txt> 25 February 1999
EX1028	U.S. Patent 6,674,713 to Berg et al.
EX1029	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Joint Submission of Proposed Amended Scheduling Order, ECF No. 57 (W.D. Tex. Mar. 4, 2021)
EX1030	Fourteenth Supplemental Order Regarding Court Operations Under the Exigent Circumstances Created By the COVID-19 Pandemic, The United States District Court for the Western District of Texas (March 17, 2021)
EX1031	<i>Digital Retail Apps, Inc. v. H-E-B, LP</i> , Case No. 6:19-cv-00167, Joint Stipulation and Order Postponing Trial, ECF No. 182 (W.D. Tex. Jan. 13, 2021)
EX1032	<i>Jenam Tech, LLC's</i> Preliminary Narrowing of Claims (October 20, 2020)
EX1033	<i>Jenam Tech, LLC's</i> Correspondence Regarding Narrowing of Claims (December 4, 2020)
EX1034	Judge Alan D. Albright's Case Statistics By Year (Retrieved from DocketNavigator on March 9, 2021)

Petition for *Inter Partes* Review
Patent No. 10,306,026

EX1035	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Order Granting Motion to Stay Case, ECF No. 58 (W.D. Tex. Mar. 10, 2021)
EX1036	U.S. Patent No. 7,389,512 to Tucker (“ <i>Tucker</i> ”)
EX1037	<i>Jenam Tech, LLC’s First Set of Infringement Contentions</i> regarding U.S. Patent No. 10,306,026 (August 21, 2020)
EX1038	U.S. Pre-Grant Publication No. 2008/0144603 to Chouksey <i>et al.</i>
EX1039	U.S. Pre-Grant Publication No. 2003/0131135 to Yun

Petition for *Inter Partes* Review
Patent No. 10,306,026

I. INTRODUCTION

Google LLC (“Petitioner”) requests *inter partes* review (“IPR”) of claims 1, 17, 21–22, 31–33, 43, 45–48, and 51 (“the challenged claims”) of U.S. Patent No. 10,306,026 (“the ’026 patent”) (EX1001) assigned to Jenam Tech, LLC (“Patent Owner” or “PO”). For the reasons below, the challenged claims should be found unpatentable and canceled.

II. MANDATORY NOTICES

Real Parties-in-Interest: Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies the following as the real parties-in-interest: Google LLC.¹

Related Matters: The ’026 patent is asserted in the following civil action: *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.). The ’026 patent was previously asserted in the following civil action: *Jenam Tech, LLC v. Samsung Electronics Co., Ltd.*, Case No. 4:20-cv-00279 (E.D. Tex.) (voluntarily dismissed).

¹Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

Petition for *Inter Partes* Review
Patent No. 10,306,026

The '026 patent claims priority to U.S. Patent Application No. 15/694,802, filed on September 3, 2017, and issued as U.S. Patent No. 9,923,995 (“the '995 patent”). (EX1001, Cover.) The '995 patent is subject to the following instituted IPRs: *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845 (“Google -845 IPR”); and *Unified Patents, LLC. v. Jenam Tech, LLC*, IPR2020-00742 (“Unified -742 IPR”).

Petitioner is concurrently filing another IPR petition challenging the '026 patent.² Petitioner is also concurrently filing IPR petitions challenging U.S. Patent No. 10,069,945 which is also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.).

Petitioner has also filed the following IPR petitions challenging patents related to the '995 and '026 patents, and which are also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.):

- IPR2021-00627 (U.S. Patent No. 10,375,215, “the '215 patent”);
- IPR2021-00628 (U.S. Patent No. 10,075,564, “the '564 patent”);
- IPR2021-00629 (the '564 patent); and
- IPR2021-00630 (U.S. Patent No. 10,075,565, “the '565 patent”).

² Petitioner concurrently submits herewith its Notice Regarding Multiple Petitions.

Petition for *Inter Partes* Review
Patent No. 10,306,026

Counsel and Service Information: Lead counsel: Naveen Modi (Reg. No. 46,224). Backup counsel: (1) Joseph E. Palys (Reg. No. 46,508), (2) Quadeer Ahmed (Reg. No. 60,835), (3) Jason Heidemann (Reg. No. 77,880).

Service information is Paul Hastings LLP, 2050 M Street NW, Washington, DC 20036, Tel.: 202.551.1700, Fax: 202.551.1705, email: PH-Google-Jenam-IPR@paulhastings.com. Petitioner consents to electronic service.

III. PAYMENT OF FEES

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-2613.

IV. GROUNDS FOR STANDING

Petitioner certifies that the '026 patent is available for review and Petitioner is not barred or estopped from requesting review on the grounds identified herein.

V. PRECISE RELIEF REQUESTED

A. Claims for Which Review Is Requested

Petitioner requests review of claims 1, 17, 21–22, 31–33, 43, 45–48, and 51 (“the challenged claims”) and cancellation of those claims as unpatentable given these grounds.

Petition for *Inter Partes* Review
Patent No. 10,306,026

B. Statutory Grounds of Challenge

Ground 1: Claims 1 and 33 are unpatentable under pre-AIA 35 U.S.C. § 102(b) over Eggert, TCP Abort Timeout Option (April 14, 2004) (“*Eggert*”) (EX1006);

Ground 2: Claims 1, 17, 21–22, 33, 43, and 45–48 are unpatentable under pre-AIA 35 U.S.C. § 103(a) over *Eggert* (EX1006) in view of U.S. Patent No. 6,981,048 to Abdolbaghian (“*Abdolbaghian*”) (EX1007);

Ground 3: Claims 31, 32, and 51 are unpatentable under pre-AIA 35 U.S.C. § 103(a) over *Eggert* and U.S. Patent No. 7,389,512 to Tucker (“*Tucker*”) (EX1036); and

Ground 4: Claims 31, 32, and 51 are unpatentable under pre-AIA 35 U.S.C. § 103(a) over *Eggert*, *Abdolbaghian*, and *Tucker* Render Obvious Claims 31–32 and 51.³

The ’026 patent issued from Application No. 16/040,517, filed July 19, 2018, which is a continuation of Application No. 15/915,047, filed March 7, 2018, which is a continuation of Application No. 15/694,802, filed September 3, 2017, which is

³ Other references discussed herein are provided to show the state of the art at the time of the alleged invention of the ’026 patent.

Petition for *Inter Partes* Review
Patent No. 10,306,026

a continuation-in-part of Application No. 14/667,642, filed March 24, 2015, which is a continuation-in-part of Application No. 13/477,402, filed May 22, 2012, which is a continuation of Application No. 12/714, 454 (“the ’454 application”), filed February 27, 2010. (EX1001, 1:8–34.)

For purposes of this proceeding only, Petitioner assumes, without conceding, the earliest effective filing date of the ’026 patent is February 27, 2010 (filing date of the ’454 application).

Eggert, an Internet Engineering Task Force (IETF) Internet-Draft (or “ID”) working document, was published on April 14, 2004. (EX1006, 1; *see also infra* Section X.) As confirmed by the declaration of Alexa Morris, Managing Director of IETF, *Eggert* was published, disseminated, and reasonably available to the public by April 15, 2004. (EX1019 ¶¶9–10; *see also infra* Section X.) Thus, *Eggert* qualifies as prior art at least under 35 U.S.C. § 102(b) (pre-AIA).

Abdolbaghian issued December 27, 2005 from an application filed November 22, 2000. (EX1007, Cover.) *Tucker* published on June 17, 2008 from an application filed January 27, 2004. (EX1036, Cover.) Therefore, *Abdolbaghian*, *Hayden*, and *Tucker* qualify as prior art at least under 35 U.S.C. § 102(b) (pre-AIA).

The above references were not considered during prosecution of the ’026 patent (EX1001, Cover (“References Cited”); *see also generally* EX1004) and are

Petition for *Inter Partes* Review
Patent No. 10,306,026

not the same as or substantially like any art previously presented to the Office. (*See infra* Section XI.A.)

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art at the time of the alleged invention of the '026 patent (“POSITA”) would have an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking. (EX1002 ¶¶16–18.)⁴ More education can supplement practical experience and vice versa. (*Id.*)

VII. OVERVIEW OF THE '026 PATENT

The '026 patent is directed to networking, and, to the sharing of information for detecting an idle TCP connection. (EX1001, 2:32–34, 8:37–46.) Figure 7 illustrates such a process.

⁴ Petitioner submits the declaration of Dr. Bill Lin (EX1002), an expert in the field of the '026 patent. (*Id.* ¶¶3–15; EX1003.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

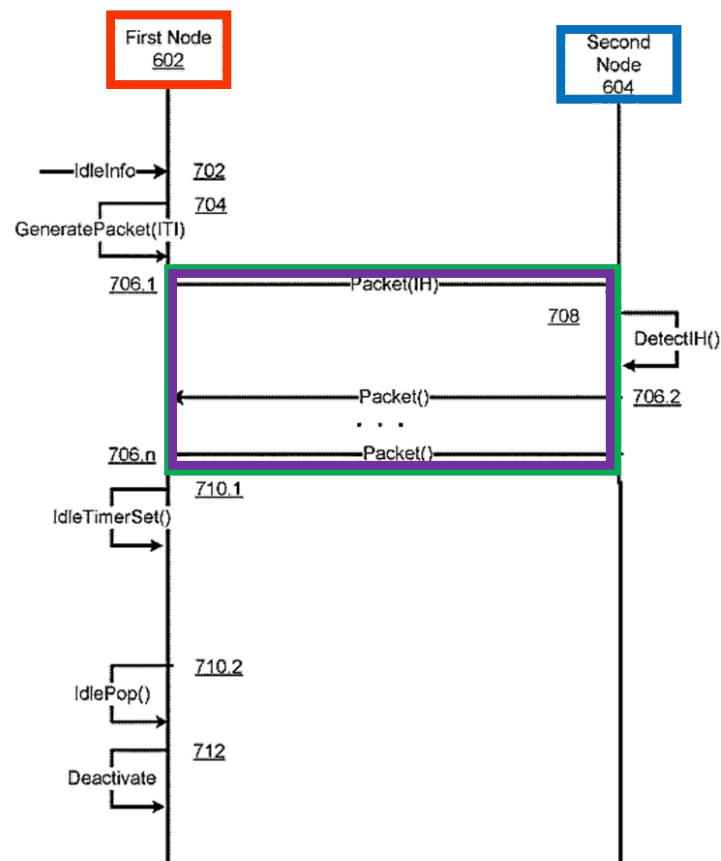


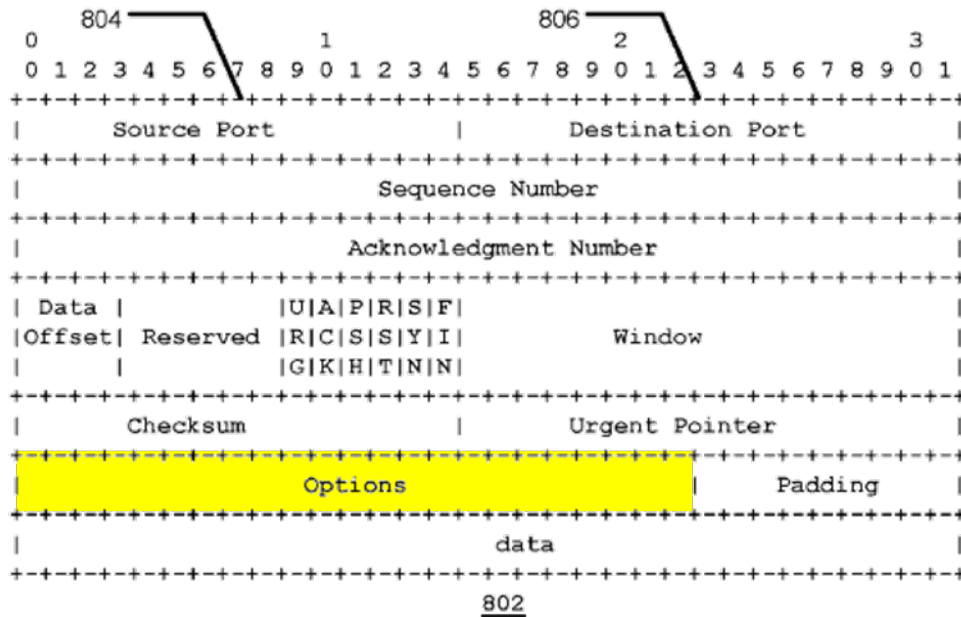
Fig. 7

(*Id.*, FIG. 7 (annotated); EX1002 ¶66.)

First, the **first node 602** receives a message 702 identifying idle information representing a duration for an idle time period (“ITP”). (EX1001, 11:62–64.) The idle information “may include and/or identify a duration of time for detecting an idle time period.” (*Id.*, 12:42–47.) The “duration may be specified according to various measures of time[,] including seconds.” (*Id.*; EX1002 ¶67.)

Petition for *Inter Partes* Review
 Patent No. 10,306,026

Next, the **first node** “generat[es] a TCP packet including an ITP header based on received idle information.” (*Id.*, 16:10–14.) The **TCP options field** of a TCP packet may store this ITP header. (*Id.*, 15:31–51.) Figure 8 below, “adapted from [IETF] RFC 793,” illustrates this configuration. (*Id.*, FIG. 8, 6:35–37, 15:31–51.)



(*Id.*, FIG. 8 (cropped and annotated); EX1002 ¶68; *see also* EX1008, FIG. 3.)

The **TCP options field** contains a KIND sub-field that identifies the type of option presented, a length sub-field that specifies the length of the option field, and an **ITP Data sub-field** containing data. (EX1001, FIG. 8.)

Petition for *Inter Partes* Review
 Patent No. 10,306,026

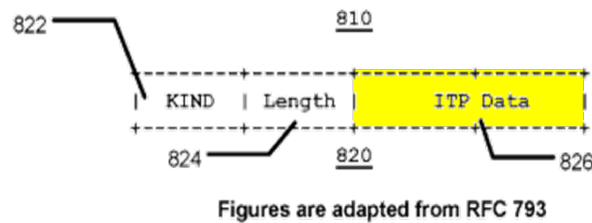


Fig. 8

(*Id.*, FIG. 8 (cropped and annotated); EX1002 ¶¶68–69.)

The ITP header is exchanged during the three-way handshake. (EX1001, 14:56–67.) For example, as shown in Figure 7, the **first node** transmits a message 706.1, i.e., a TCP packet including an ITP header (IH) that contains ITP information, to **second node 604**. (*Id.*, 16:35–36.) Message 708 exemplifies the **second node**'s detection of the IH in the received TCP packet. (*Id.*, 21:58–63; EX1002 ¶70.)

The '026 patent explains that “a TCP keep-alive option, a TCP user timeout, a retransmission timeout, an acknowledgment timeout, and/or another timeout associated with a TCP connection may be modified based on the first idle information.” (EX1001, 13:54–57; EX1002 ¶71.)

The challenged claims recite limitations relating to features discussed above. However, all the limitations in the challenged claims were known in the prior art and obvious. (*See* Section IX; EX1002 ¶72; *see also id.* ¶¶19–65 (discussing technology background), citing, e.g., EXS. 1008–1009, 1011–1015, 1020–1028, 1036.)

VIII. CLAIM CONSTRUCTION

Under the applicable standard in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), claim terms are typically given their ordinary and customary meanings as understood by a POSITA at the time of the invention based on the claim language, specification, and the prosecution history. *Phillips*, 415 F.3d at 1313; *see also id.* at 1312–16. The Board, however, only construes the claims when necessary to resolve the controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper No. 11 at 16 (Aug. 14, 2015) (citing *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)). Petitioner believes that no express constructions of any claim terms are necessary to assess whether the prior art reads on the challenged claims.⁵ (EX1002 ¶73.)

⁵ Petitioner reserves all rights to raise claim construction and other arguments, including challenges under 35 U.S.C. §§ 101 or 112, in district court as relevant to those proceedings. *See, e.g., Target Corp. v. Proxicom Wireless, LLC*, IPR2020-00904, Paper 11 at 11–13 (November 10, 2020). A comparison of the claims to any accused products in litigation may raise controversies that are not presented here given the similarities between the references and the patent.

Petition for *Inter Partes* Review
Patent No. 10,306,026

Challenged claims 1 and 33 recite the terms “TCP-variant packet” and “TCP-variant connection” (EX1001, 24:21–28, 26:17–30), which are also recited in claims of the related ’995 patent at-issue in the Unified -742 IPR and Google -845 IPR (*see supra* Section II). The ’026 patent and the ’995 patent share a common specification and drawings. (*Compare* EX1001 *with* EX1010 (’995 patent).)

The ’026 patent discloses that the “TCP-variant” protocol varies slightly from TCP by adding an idle timeout option using the format for new TCP options in a TCP header. (EX1001, 15:31–60, 16:10–14, FIG. 8; *supra* Section VII.)

Like the Board noted for the ’995 patent, the ’026 patent’s specification “provides little guidance as to [the] meaning [of ‘TCP-variant connection’].” Google -845 IPR, Paper 16 at 17 (October 8, 2020); Unified -742 IPR, Paper 11 at 5 (October 8, 2020). (*See generally* EX1001.)⁶

Eggert’s timeout option is a similar variation on TCP as disclosed in the ’026 patent and thus discloses the term “TCP-variant.” (*See, e.g., infra* Sections IX.A.1.b–d.) As shown below, *Eggert*’s timeout option (right) uses the same TCP

⁶ While the ’026 patent generically refers to “variant of the current TCP” (EX1001, 16:66–67), the term “TCP-variant” is only found in the claims.

Petition for *Inter Partes* Review
 Patent No. 10,306,026

structure as the ITP field illustrated in Figure 8 of the '026 patent (left), including a data portion (YELLOW) containing the timeout value.

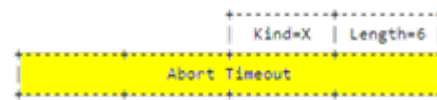
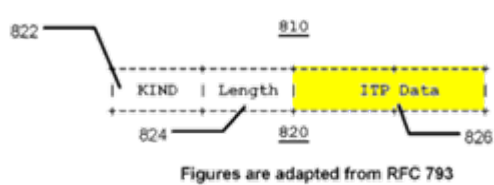


Figure 1: TCP Abort Timeout Option

EX1001, FIG. 8 (annotated)

EX1006, FIG. 1 (annotated)

(EX1002 ¶¶83, 109, 173–174; *infra* Section IX.A.1.b–d.) Moreover, the Board preliminarily found that *Eggert* discloses these TCP-variant limitations in the Google -845 IPR. Google -845 IPR, Paper 16 at 20–23.

Thus, given that the '026 patent offers no descriptions that would require a special meaning of these terms and that the prior art discloses these features under any reasonable interpretation, the Board need not construe the TCP-variant terms. Instead, as Petitioner does, the Board should apply the plain meaning of the terms “TCP-variant packet” and “TCP-variant connection” like the Board has applied in the Unified -742 IPR and Google -845 IPR. (*See infra* Sections IX.A.1.b–e, IX.A.2.b–e.)

IX. DETAILED EXPLANATION OF GROUNDS**A. Ground 1: *Eggert* Anticipates Claims 1 and 33****1. Claim 1****a) A method, comprising:**

To the extent that the preamble is limiting, *Eggert* discloses the limitations therein. (EX1002 ¶¶94–100.) For example, *Eggert* discloses features relating to modifying a TCP implementation (e.g., a TCP-variant protocol) on a host to support “a new TCP option - the Abort Timeout Option - that allows conforming hosts to negotiate per-connection abort timeouts.” (EX1006, 3.) The hosts (e.g., mobile devices) that use *Eggert*’s TCP-variant protocol perform steps to negotiate a per-connection abort timeout that allows those “hosts to maintain TCP connections across disconnected periods that are longer than their system’s default abort timeout.” (*Id.*, 3; *see also id.*, Abstract, 5–7 (describing “[o]peration” of the TCP Abort Timeout Option (ATO) by host(s).)⁷ *Eggert* thus discloses that the host(s) would perform a process to negotiate an abort timeout for the connection when implementing *Eggert*’s disclosed processes. (EX1002 ¶¶95–100, 74–83 (overview of *Eggert*); *see also infra* Sections IX.A.1.b–e.)

⁷ *Eggert* uses “hosts” and “nodes” interchangeably. (EX1006, 5 (“the node receiving the SYN”), 7 (“the receiving host”); EX1002 ¶98 n.3.)

- b) **at a node: receiving, from another node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established;**

Eggert discloses this limitation. (EX1002 ¶¶101–111.) *Eggert* discloses this limitation in two different ways such that an **initiator** (“node”) receives a “SYN/ACK+ATO” segment (“TCP-variant packet”) from the **responder** (“another node”) during a three-way handshake, which precedes the establishment of the TCP-variant connection. (EX1002 ¶102.)⁸

For example, *Eggert*’s TCP-variant protocol uses the three-way handshake to establish a TCP-variant connection between two nodes on a network. (EX1006, 5.) A three-way handshake involves the exchange of synchronize (SYN) and acknowledge (ACK) messages between the nodes to create a connection—i.e., SYN, SYN-ACK, and ACK messages in a standard TCP implementation. (EX1002 ¶103;

⁸ The ’026 patent describes a “packet” in a manner consistent with what the TCP standard (RFC 793) describes as “segment.” (*Compare* EX1001, 14:56–15:3, FIG. 8 *with* EX1008, 15, FIG. 3 (showing the same TCP header in a TCP “packet” (EX1001) and TCP “segment” (EX1008)).) Thus, *Eggert*’s TCP “segment” is the same as a TCP “packet” as claimed. (EX1002 ¶102.)

EX1008, 31–32, 34–37; EX1001, 13:49–63 ('026 patent disclosing the standard TCP three-way handshake).) The connection is not established until after the third ACK message in the handshake. (EX1002 ¶103.)

As discussed below and exemplified in Figure 2 below, *Eggert* discloses two ways that the **initiator** (“node”) receives a “SYN/ACK+ATO” segment (“TCP-variant packet”) from the **responder** (“another node”) during the three-way handshake in advance of a TCP-variant connection being established. (EX1006, 5.)

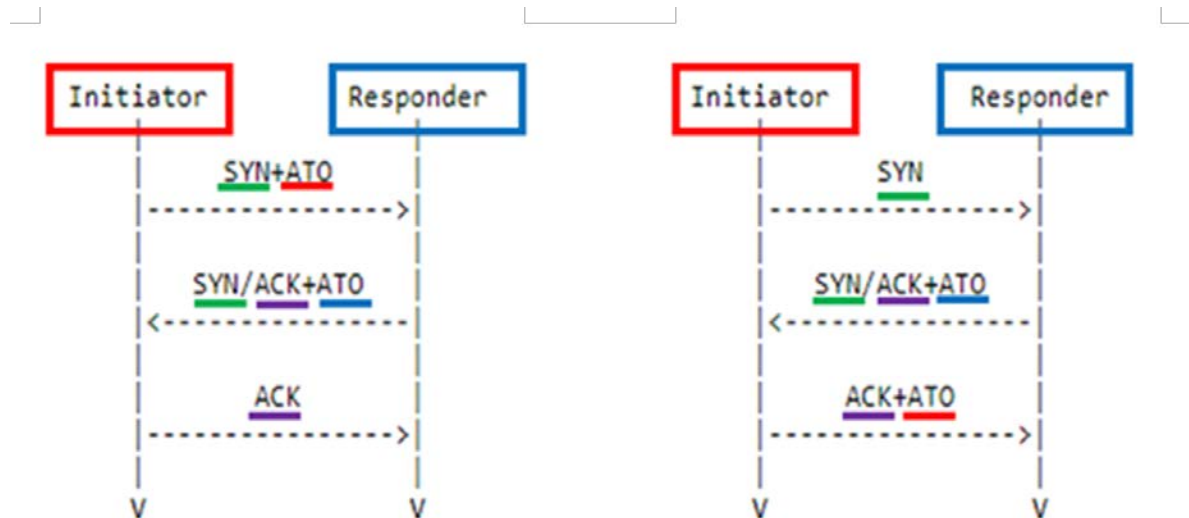


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

(EX1006, FIG. 2 (annotated); EX1002 ¶104.)

First, as shown on the left-hand side of Figure 2 (when the **initiator** starts an abort timeout negotiation), the **initiator** sends the **responder** a “SYN+ATO” segment. (EX1006, 5–7, FIG. 2.) The ATO in the segment reflects that this segment

Petition for *Inter Partes* Review
Patent No. 10,306,026

contains an abort timeout. (*Id.*, 3.) So when the **responder** accepts/modifies the **initiator**'s offered abort timeout value, it must include it in the ATO it sends. (*Id.*, 5–7.) The **initiator** thus receives a “SYN/ACK+ATO” segment (“TCP-variant packet”) from the **responder** before the TCP-variant connection is established. (*Id.*, EX1002 ¶¶105–106.)

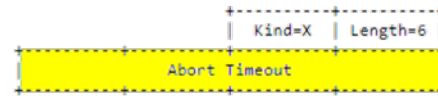
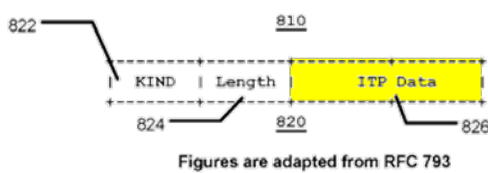
Second, as shown on the right-hand side of Figure 2 (when the **responder** starts an abort timeout negotiation), the **initiator** also receives an “SYN/ACK+ATO” segment (“TCP-variant packet”) from a **responder** before the TCP-variant connection is established. (EX1006, 5–7; EX1002 ¶107.)

The received “SYN/ACK+ATO” segment in both scenarios is a TCP-variant packet because it contains an abort timeout option (ATO), an option not standard in the TCP protocol defined by RFC 793. (EX1002 ¶108.) *See also* Google -845 IPR, Paper 16 at 20–21 (Board agreeing that *Eggert* discloses a TCP-variant packet for similar reasons).

Indeed, *Eggert*'s TCP ATO shown in Figure 1 uses the same TCP structure as the idle time period (ITP) field illustrated in Figure 8 of the '026 patent. For example, like the '026 patent, *Eggert*'s ATO is also incorporated in the TCP header and arranged in a similar format. (*Compare* EX1001, 15:52–60, FIG. 8 with EX1006, 3–5, FIG. 1.) As shown below, both formats use a KIND sub-field to

Petition for *Inter Partes* Review
 Patent No. 10,306,026

indicate the new option, a TCP option length sub-field, and a **field** of the TCP option containing the timeout value. (*Compare* EX1001, FIG. 8 *with* EX1006, FIG. 1.)



(EX1001, FIG. 8 (annotated) (left); EX1006, FIG. 1 (annotated) (right); EX1002 ¶109.) Thus, *Eggert* uses the same format for the ATO as the embodiment disclosed in the '026 patent for a TCP-variant packet. (EX1002 ¶109.)

The understanding that *Eggert*'s "SYN/ACK+ATO" segment is a TCP-variant packet in the context of the '026 patent also tracks PO's proposed construction of "variant" in a related matter. (*See, e.g.*, EX1018, 60 (Exhibit C) (PO defining a variant as one that "manifest[s] variety, deviation, or disagreement, [or] var[ies] slightly from the standard.")) *Eggert* describes a TCP-variant packet under PO's interpretation because the packets exchanged during the three-way handshake in *Eggert* vary, as described above, from those exchanged in the TCP standard.

Thus, a connection using *Eggert*'s ATO would be a TCP-variant connection, and at least the segments using the TCP ATO for establishing such a TCP-variant connection would be TCP-variant packets. (EX1002 ¶109; *supra* Section VIII.)

Accordingly, *Eggert* discloses limitation 1.b. (EX1002 ¶111.)

c) **detecting an idle time period parameter field in the TCP-variant packet;**

Eggert discloses this limitation. (EX1002 ¶¶112–120.) *Eggert* explains that the **initiator** detects the ATO parameter field (“idle time period parameter field”) in the “SYN/ACK+ATO” segment (“TCP-variant packet”).

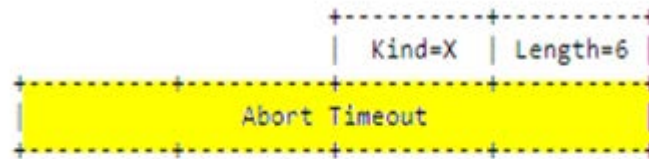


Figure 1: TCP Abort Timeout Option

(EX1006, FIG. 1 (annotated); EX1002 ¶¶113–114; *see also supra* Section IX.A.1.b.)

First, regarding the left-hand side of Figure 2, *Eggert* explains the **initiator** must use the **responder**’s returned abort timeout value for the connection because the **responder** has the ability to accept or shorten the abort timeout value that was proposed by the **initiator** in the initial “SYN+ATO” segment it sent to the **responder**. (EX1006, 5–7.)

Second, regarding the right-hand side of Figure 2, the **initiator** may either accept or shorten the offered abort timeout from the **responder**. (*Id.*)

Thus, for either side of Figure 2, the **initiator** must first detect the ATO parameter field (“idle time period parameter field”) in the “SYN/ACK+ATO”

Petition for *Inter Partes* Review
Patent No. 10,306,026

segment containing the ATO value before it can use, accept or shorten the value. (EX1006, 5.) Thus, *Eggert* discloses detecting an ATO field in the TCP-variant packet. (EX1002 ¶¶116–117.)

The ATO parameter field represents an idle time period parameter field used to convey an abort timeout value between two hosts. (EX1006, 1–7, FIG. 1.) The ATO field contains the abort timeout value, which defines an abort timeout (“idle time period”), as claimed. (*Id.*, 2–5, FIG. 1.) For example, the abort timeout value in the ATO field “is the desired abort timeout of the connection, specified in seconds.” (*Id.*, 3, FIG. 1; *see also id.*, Abstract.) The abort timeout value defines an idle time period because it is associated with a period where no packet is communicated in the connection to keep the connection active. (EX1002 ¶¶118–119; EX1006, Abstract; *see infra* Section IX.A.1.d.)

Accordingly, *Eggert* discloses limitation 1.c. (EX1002 ¶120.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

- d) **identifying metadata in the idle time period parameter field for an idle time period and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active; and**

Eggert discloses this limitation. (EX1002 ¶¶121–134.) *Eggert* discloses that the **initiator** identifies an abort timeout value (“metadata”) in the ATO parameter field (“idle time period parameter field”) for an abort timeout (“idle time period”), during which no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active. (*See also supra* Section IX.A.1.c.)

For example, as discussed above, *Eggert* explains that the **initiator** accepts or shortens the offered abort timeout value within the ATO parameter field of the “SYN/ACK+ATO” segment. (*Supra* Section IX.A.1.c; EX1006, 5–7.) Thus, the **initiator** must first identify the abort timeout value (“metadata”) before accepting or shortening the offered abort timeout. (EX1006, 5–7.) Consequently, *Eggert* discloses that the **initiator** processes the received “SYN/ACK+ATO” segment according to an established format (*Id.*, FIG. 1) to identify which bits of the option represent the proposed abort timeout value. (*Id.*, 3–5; EX1002 ¶¶122–123.)

Eggert explains that the “Abort Timeout” is “specified in seconds” (EX1006, 3) and is represented by a “32-bit value” (*id.*, 9). Thus the bits that identify the ATO field (e.g., KIND, LENGTH) and the bits representing the abort timeout value each

Petition for *Inter Partes* Review
Patent No. 10,306,026

constitute “metadata” for the abort timeout (“idle time period”). (EX1002 ¶124.) Further, as discussed above for claim limitation 1.c, the format for representing the TCP abort timeout in Figure 1 of *Eggert* is like the ITP data field 826 in Figure 8 of the ’026 patent. (*Id.*; *supra* Section IX.A.1.c.)

Eggert’s disclosures for identifying such values are consistent with the ’026 patent, which explains that the metadata can be a “duration of time” (EX1001, 22:2–5), which is what the ATO value in *Eggert* contains (EX1006, 3–5, FIG. 1.) (EX1002 ¶125.) Moreover, this understanding is consistent with PO’s view of the same claimed features. (EX1017, 8 (PO alleging “identifying metadata (e.g., a value, etc.) in the idle time period parameter field”), (“the metadata includes a value in seconds”).)

Eggert further discloses that the timeout value in the ATO field corresponds to an idle time period, “during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active,” as claimed. (EX1002 ¶126.) For example, *Eggert* explains that the abort timeout duration specified in the ATO field is used to track periods during which there is no communication to keep the connection active. (EX1006, 1–7; *see also id.*, 9 (“Long abort timeout values allow hosts to tolerate extended periods of disconnection”).) The packet that must be “communicated . . . to keep the . . . connection active” in *Eggert* is an ACK that a

Petition for *Inter Partes* Review
Patent No. 10,306,026

node receives in response to a previously sent segment. (EX1002 ¶127.) Indeed, *Eggert*'s ATO relies upon "[t]he TCP specification [1] [which] includes a 'user timeout' that defines the maximum amount of time that segments may remain *unacknowledged* before TCP will abort the connection. If a disconnection lasts longer than the user timeout, the TCP connection will abort." (EX1006, 3 (emphasis added).) Like *Eggert*, the '026 patent also describes modifying a TCP user timeout. (*Compare* EX1001, 13:54–57, 22:23–30 *with* EX1006, 3.) Accordingly, *Eggert*'s description of the abort timeout as the duration of time during which a node can wait to receive an ACK is a period "during which no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active." (EX1002 ¶¶128–129.)

(1) Disconnection

Eggert also discloses that the timeout value in the ATO field corresponds to an idle time period, "during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active," as claimed, in an **additional**

Petition for *Inter Partes* Review
Patent No. 10,306,026

way where a disconnection occurs between **initiator** and **responder**.⁹ (EX1006, 1–3; EX1002 ¶¶130–133.)

For example, when an interruption in communication occurs, such as a disconnection in the network service (as contemplated by *Eggert*) or in a physical network medium (as described in the '026 patent) (EX1001, 2:11–14, 2:28–31, 12:67–13:6), packets from one node do not reach the other node in either direction. (EX1002 ¶¶131–132.) In this situation contemplated by the '026 patent, no packets are communicated between the nodes to keep the connection active because they cannot get through the interruption or the physical medium's disconnection. (EX1002 ¶132; EX1001, 12:64–13:6, 2:28–31, 20:36–45, FIG. 7 (annotated below in yellow showing no communication between the nodes during a disconnection).)¹⁰

⁹ This tracks how the '026 patent describes such features. (EX1001, 2:28–31, 12:64–13:6, 20:36–45 (describing that the idle time period may be linked to the physical disconnection of a network medium or simply a dead connection).)

¹⁰ The '026 patent confirms that in order for a packet to be “communicated . . . to keep the . . . connection active,” a packet sent by one node must be received by the other node. (EX1002 ¶133; EX1001, 11:41–46 (“With reference to the method illustrated in FIG. 2, block 202 illustrates the method includes receiving, by a first

Petition for *Inter Partes* Review
Patent No. 10,306,026

Indeed, neither node would have received any packets from the other node during such a disconnection period. Therefore, neither node would have attempted to send a responsive acknowledgment packet during the period. (EX1002 ¶132.) This is the type of situation where, as discussed above, *Eggert* aims to maintain the connection, e.g., maintain the connection even when no packet is communicated during the negotiated abort timeout period, as there would be no received packets to acknowledge. *Power Integrations, Inc. v. Fairchild Semiconductor Int'l, Inc.*, 843 F.3d 1315, 1336-37 (Fed. Cir. 2016) (finding claim feature anticipated where prior art met the feature sometimes during operation); *Hewlett-Packard Co. v. Mustek Sys., Inc.*, 340 F.3d 1314, 1326 (Fed. Cir. 2003) (“Just as ‘an accused product that

node, first idle information for detecting a **first idle time period during which no TCP packet** including data in a first data stream **sent** in the TCP connection **by a second node** is *received by the first node.*”), 11:46–51 (disclosing that during the idle time period, no TCP packet “**sent in the TCP connection by a second node** is *received by the first node*”), 11:51–57 (same), 21:18–37 (same), 19:64–20:6 (disclosing that during the idle time period, “**no TCP packet in the TCP connection is received**, by the first node 602 . . . from second node 604”), FIG. 2 (block 202), FIG. 3 (block 304), FIGS. 6–7.)

Petition for *Inter Partes* Review
 Patent No. 10,306,026

sometimes, but not always, embodies a claimed method nonetheless infringes,' a prior art product that sometimes, but not always, embodies a claimed method nonetheless teaches that aspect of the invention") (quoting *Bell Commc 'ns Research, Inc. v. Vitalink Commc 'ns Corp.*, 55 F.3d 615, 622-623 (Fed. Circ. 1995)).

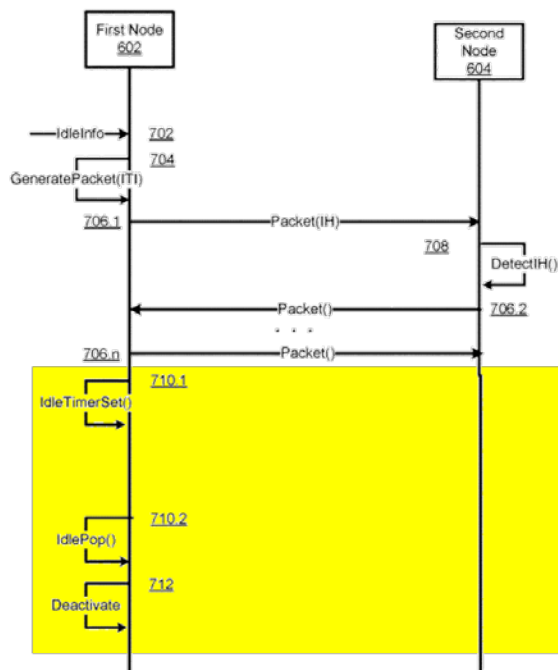


Fig. 7

(EX1001, FIG. 7 (annotated); EX1002 ¶132.)

* * *

Thus, *Eggert* discloses limitation 1.d. (EX1002 ¶134.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

- e) **modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.**

Eggert discloses this limitation. (EX1002 ¶¶135–141.) *Eggert* discloses modifying, based on the abort timeout value (“metadata”), a timeout attribute associated with the TCP-variant connection in at least two ways. (*See also supra* Sections IX.A.1.b–d.)

For instance, *Eggert* discloses that while “[m]any TCP implementations default to user timeout values of a few minutes” (EX1006, 3), after successful negotiation, hosts use the abort timeout for the connection (*id.*, 7). Figure 2 discloses the two scenarios for ATO negotiation, both of which disclose claim limitation 1.e.

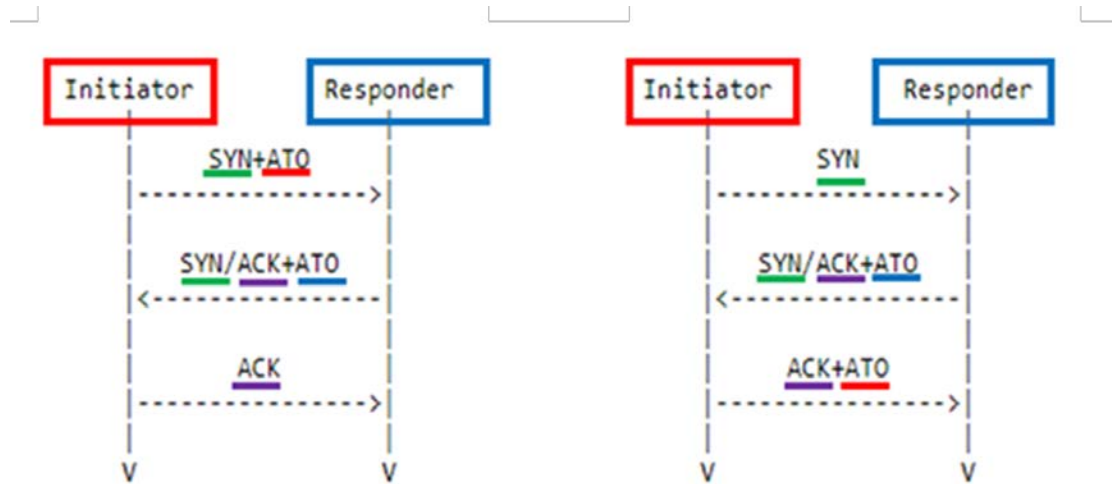


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

(*Id.*, FIG. 2 (annotated); EX1002 ¶137.)

First, the **initiator** in the left-hand side scenario of Figure 2 has only a single option because it proposed the abort timeout value. (EX1006, 5–7.) The **initiator** “MUST be prepared to accept a shorter timeout value [from the **responder**] than proposed after the negotiation.” (*Id.*, 5.) The **initiator**’s acceptance of a shorter timeout value from the **responder** for the TCP-variant connection describes a “modifying” process like that recited in limitation 1.e. (EX1002 ¶138.)

Second, as shown on the right-hand side of Figure 2, the **initiator**, because the **responder** specified the initial ATO in the “SYN/ACK+ATO” segment, may decide “whether to accept, *shorten*, or reject its peer’s proposed abort timeout.” (EX1006, 5–7; *see also id.*, 9.) Thus, the **initiator** modifies a timeout attribute, the

Petition for *Inter Partes* Review
Patent No. 10,306,026

abort timeout for the connection, based on shortening the value (“metadata”) in the ATO field. (EX1002 ¶139.)

The ’026 patent describes timeout attributes broadly and matches *Eggert*’s abort timeout value. (EX1001, 22:23–28 (“ITP option handler component 562 may ... modify one or more corresponding attributes of a keep-alive option, a TCP user timeout, a retransmission timeout, an acknowledgment timeout, and another timeout associated with the TCP connection, in response to identifying the ITP header”); EX1002 ¶140.)

Thus, *Eggert* discloses limitation 1.e. (EX1002 ¶141.)

2. Claim 33

a) A method, comprising:

To the extent limiting, *Eggert* discloses the features of this preamble for the same reasons discussed above for limitation 1.a. (*See* Section IX.A.1.a; *see also* EX1002 ¶¶142–143; *infra* Sections IX.A.2.b–e.)

b) at a node: receiving idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active;

Eggert discloses this limitation. (EX1002 ¶¶144–167.) As explained below, the **initiator** (“node”) receives an abort timeout value (“idle information”) for

Petition for *Inter Partes* Review
 Patent No. 10,306,026

detecting an idle time period (a period, during which, no packet is communicated in the connection to keep the connection active).

As explained above for limitation 1.b, *Eggert* discloses a TCP variant protocol. (*Supra* Section IX.A.1.b.) As discussed and exemplified in Figure 2 below, the **initiator** receives this abort timeout value in two different ways.

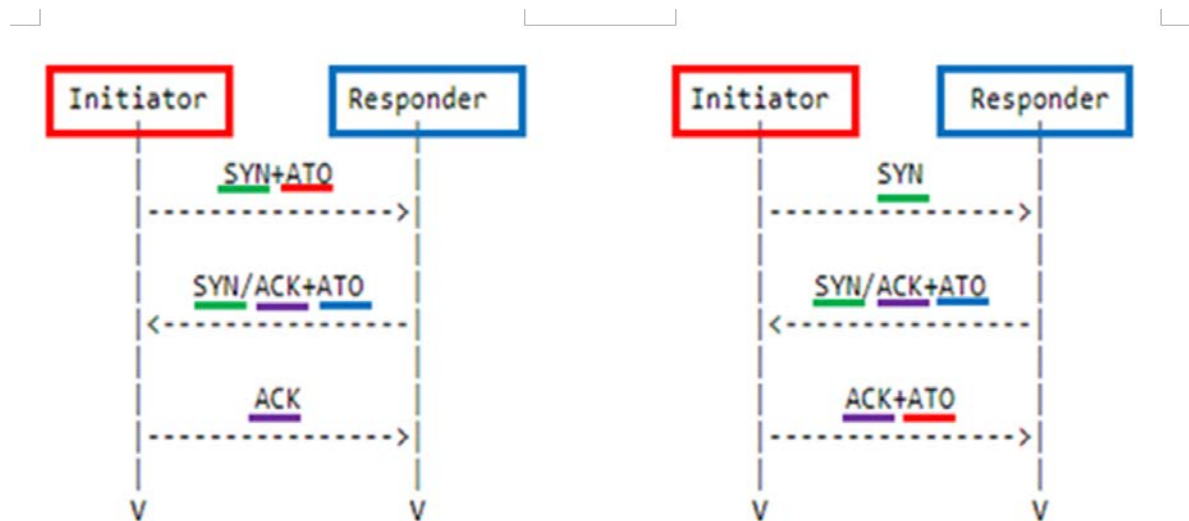


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

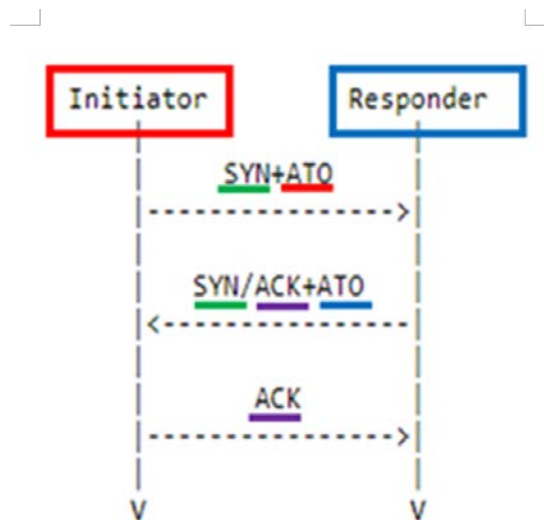
(EX1006, FIG. 2 (annotated); EX1002 ¶¶146–147.)

First, when an **initiator** initiates an abort timeout negotiation, the **initiator** must receive the proposed abort timeout value (“idle information”), to include in a “**SYN+ATO**” segment, for detecting an abort timeout period, as explained below in Section IX.A.2.b.1. Second, when the **responder** initiates an abort timeout negotiation, the **initiator** must receive the proposed abort timeout value in the

received “SYN/ACK+ATO” segment, for detecting an abort timeout period, as explained below in Section IX.A.2.b.2. Under both circumstances, when there is a disconnection preventing communication between the two nodes as envisioned by *Eggert*, *Eggert* also discloses this feature in an additional way, as explained in Section IX.A.2.b.3. (EX1002 ¶148.)

(1) “SYN+ATO”

First, regarding the left-hand side of Figure 2 (when the **initiator** initiates an abort timeout negotiation), the **initiator** sends the **responder** a “SYN+ATO” segment. (EX1006, 5–7, FIG. 2.)



(*Id.*, FIG. 2 (cropped and annotated); EX1002 ¶149.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

As explained above, the ATO contains an **Abort Timeout** sub-field containing the timeout value for detecting an abort timeout period (“idle time period”) between hosts. (*Supra* Section IX.A.1.b; EX1002 ¶¶150–151)

When the **initiator** initiates the abort timeout negotiation, the **initiator** must receive an abort timeout value (“idle information”) to include in the “**Abort Timeout**” sub-field in the ATO. (EX1006, 5, 9.) Therefore, the **initiator** necessarily receives the abort timeout value before sending a “SYN+ATO” segment with the proposed abort timeout value. (EX1002 ¶152.) For example, the **initiator** may receive the abort timeout value in the form of a default timeout (EX1006, 3, 7, 9) or as a bound for the negotiated abort timeout (*id.*, 11). These values would be data values stored in memory, which are retrieved (e.g., “received”) from memory when the **initiator** generates the “SYN+ATO” segment. (EX1002 ¶¶153–155)

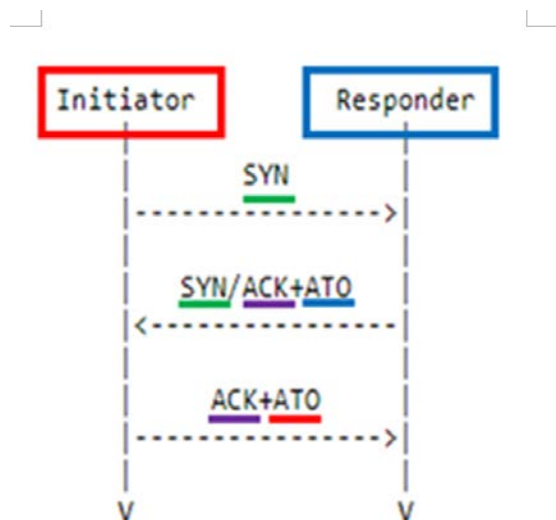
Eggert’s disclosures regarding receiving such values are consistent with the ’026 patent’s teachings, which explains that the idle information may be received (e.g., retrieved) from memory. (EX1001, 12:4–12:6; EX1002 ¶156.) Moreover, this understanding follows PO’s view of the same claimed features. (EX1017, 29–30 (alleging that the idle timeout value, specified in milliseconds, in the accused product corresponds to “idle information”).)

Eggert further discloses that the timeout value (“idle information”) corresponds to an idle time period, “during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active,” for the same reasons as explained above in relation to limitation 1.d. (*Supra* Section IX.A.1.d; EX1002 ¶¶157–159.)

Thus, *Eggert* discloses limitation 33.b in this first way. (EX1002 ¶160.)

(2) “SYN/ACK+ATO”

Regarding the right-hand side of Figure 2 (when the **responder** initiates an abort timeout negotiation), the **initiator** receives a “SYN/ACK+ATO” segment from a **responder**. (EX1006, 5–7.) The ATO identifies that this segment contains an Abort Timeout Option with a proposed abort timeout for the connection. (*Id.*, 3, 5.)



(EX1006, FIG. 2 (annotated); EX1002 ¶¶161–162.)

Thus, the **initiator** receives the proposed abort timeout value (“idle information”) for detecting an abort timeout period (“idle time period”) in the received “SYN/ACK+ATO” segment (e.g., to accept, shorten, or reject this value as discussed further below with respect to limitations 33.c–e.). (EX1002 ¶162.)

Eggert’s disclosures regarding receiving this information follows the ’026 patent, which provides that a node can receive idle information from “a message received via a network” (EX1001, 11:62–12:3) or “identified in a packet [with an ITP header] in the TCP connection” (*id.*, 12:32–33). (EX1002 ¶163.)

Like above for the “SYN+ATO” segment, the abort timeout value in the received “SYN/ACK+ATO” segment corresponds to an idle time period, “during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active,” as claimed. (*Supra* Section IX.A.1.d; EX1002 ¶164.)

Thus, *Eggert* discloses limitation 33.b in this second way. (EX1002 ¶165.)

(3) Disconnection

Besides the two ways explained above, the features of this limitation are **also** disclosed by *Eggert*, where a disconnection occurs between **initiator** and **responder**

for the same reasons as discussed above for limitation 1.d. (*See supra* Section IX.A.1.d.(1); EX1002 ¶¶166–167.)

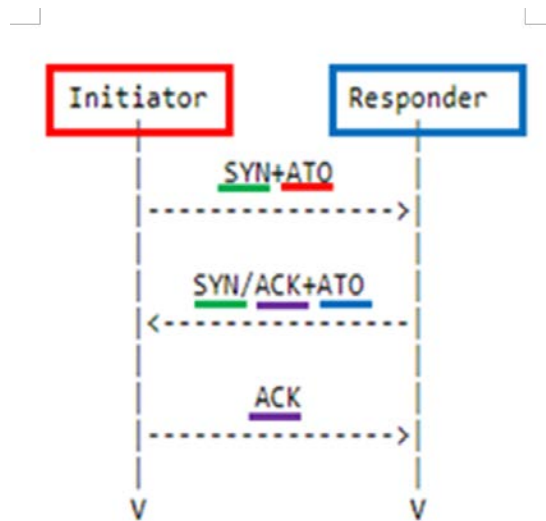
- c) **generating a TCP-variant packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information; and**

Eggert discloses this limitation. (EX1002 ¶¶168–181.) *Eggert* discloses at least two scenarios where the **initiator** generates a segment (“TCP-variant packet”) that includes an ATO parameter field (“idle time period parameter field”) identifying an abort timeout value (“metadata”) for the abort timeout (“idle time period”) based on the abort timeout value (“idle information”). (*Supra* Section IX.A.2.b; EX1006, 3, 5–7, FIGS. 1, 2.)

(1) **“SYN+ATO”**

Regarding the left-hand side of Figure 2, the **initiator** generates a “**SYN+ATO**” segment (“a TCP-variant packet”) that contains the ATO according to an established format to send to the **responder** during a three-way handshake. (EX1006, FIG. 2; *see also id.*, 5–7.)

Petition for *Inter Partes* Review
 Patent No. 10,306,026



(*Id.*, FIG. 2 (annotated); EX1002 ¶¶170.)

The ATO incorporated in the generated “SYN+ATO” segment comprises three-parameter fields identifying metadata for the idle time period, as shown in Figure 1 (below): a Kind, Length, **Abort Timeout** field. (See EX1006, 3, FIG. 1.)

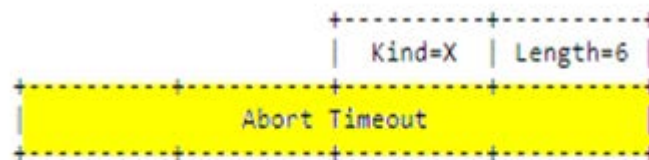


Figure 1: TCP Abort Timeout Option

(EX1006, FIG. 1 (annotated); EX1002 ¶¶171-173; *see also supra* Sections IX.A.1.b–c.) .)

The **Abort Timeout** field contains the proposed abort timeout value (“metadata”), represented by a “32-bit value,” for the abort timeout (“idle time

Petition for *Inter Partes* Review
Patent No. 10,306,026

period”), specified in seconds, based on the received abort timeout value (“idle information”) (e.g., received from memory or over the network as explained above). (EX1006, 3–5; *supra* Sections IX.A.2.b, IX.A.1.d.) The data in the other fields—KIND, LENGTH—each also constitutes “metadata” for the proposed abort timeout that are based on the abort timeout value. (EX1002 ¶¶171–172.)

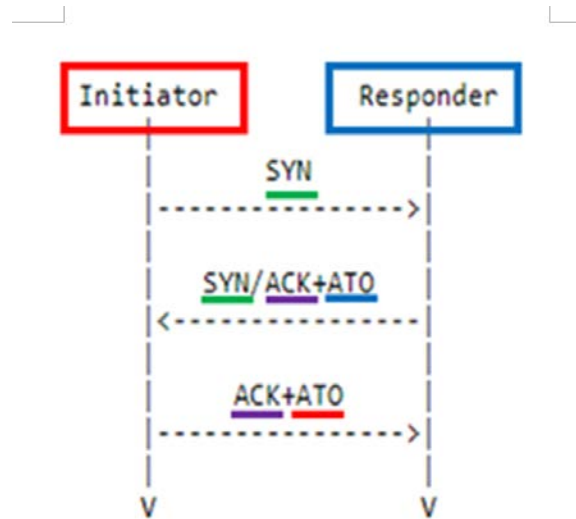
The “SYN+ATO” segment is a TCP-variant packet for at least the same reasons discussed above for claim 1. (EX1002 ¶¶173–174; *supra* Sections IX.A.1.b–c.)

Eggert thus discloses limitation 33.c in this first way. (EX1002 ¶175.)

(2) “ACK+ATO”

Regarding the right-hand side of Figure 2, the **initiator** generates an “ACK+ATO” segment to send to the **responder** during the three-way handshake. (EX1006, 5–7, FIG. 2.)

Petition for *Inter Partes* Review
 Patent No. 10,306,026



(*Id.*, FIG. 2 (annotated); EX1002 ¶176.)

Like above, the ATO incorporated in the generated “ACK+ATO” segment (“TCP-variant packet”) comprises three fields, the Kind, Length, and **Abort Timeout** fields, and each constitutes metadata for the abort timeout (“idle time period based on the idle information”). (EX1006, 3–5, FIG. 1; *supra* Sections IX.A.1.b, IX.A.2.c.1; EX1002 ¶¶177–178)

Moreover, the “ACK+ATO” segment is a TCP-variant packet for at least the same reasons discussed above for claim 1. (EX1002 ¶179; *supra* Sections IX.A.1.b–c.)

Eggert thus discloses limitation 33.c in this second way. (EX1002 ¶180.)

* * *

Petition for *Inter Partes* Review
Patent No. 10,306,026

Eggert's disclosures regarding identifying such values tracks the '026 patent, which explains that the metadata can be a "duration of time" (EX1001, 16:1–9, 22:2–6), which is what *Eggert*'s ATO value contains (EX1006, 3–5, FIG. 1). (EX1002 ¶181.) Moreover, this understanding follows PO's view of the same claimed features. (EX1017, 30 (disclosing that the accused product "identif[ies] metadata (e.g., a value, etc.) for the idle time period based on the idle information").)

Eggert thus discloses limitation 33.c in the two ways noted above. (EX1002 ¶181.)

- d) **sending, from the node to another node, the TCP-variant packet in advance of the TCP-variant connection being established to provide the metadata for the idle time period to the another node,**

Eggert discloses this limitation. (EX1002 ¶¶182–188.) As explained below, *Eggert* discloses in at least two ways that the **initiator** sends a TCP-variant packet during a three-way handshake, in advance of the establishment of the TCP-variant connection, to the **responder** ("another node") to provide the abort timeout value

(“metadata”) to the **responder** for negotiating an abort timeout for the TCP-variant connection.

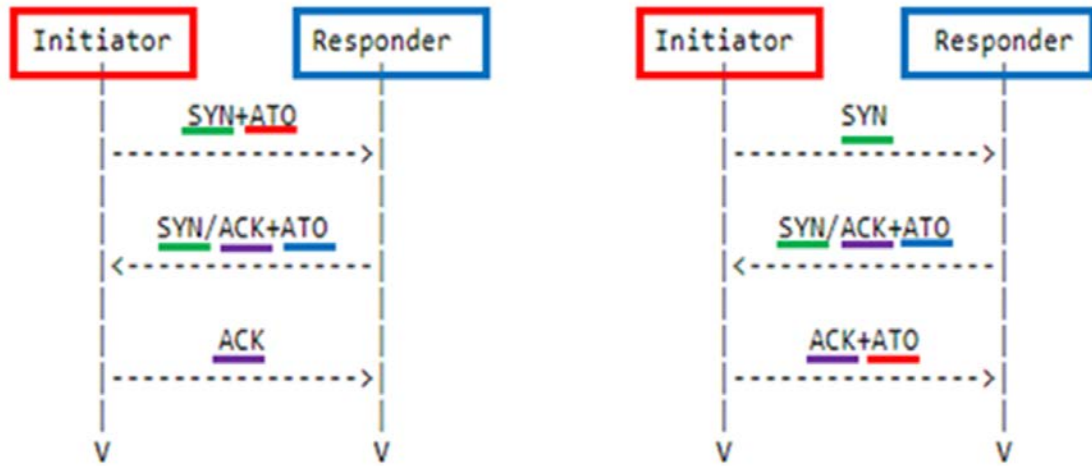


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

(EX1006, FIG. 2 (annotated); EX1002 ¶183.)

(1) “SYN+ATO”

Regarding the left-hand side of Figure 2, the **initiator** sends a “SYN+ATO” segment to the **responder**. (EX1006, FIG. 2; *see also id.*, 5–7; *supra* Section IX.A.2.c.1.) Upon receipt of this segment, the **responder** decides whether to accept, shorten, or reject the **initiator**’s proposed abort timeout. (EX1006, 5; *see also id.*, 9.) Thus, the **initiator** sends the “SYN+ATO” segment to the **responder** to provide the proposed abort timeout value for the abort timeout of the connection. (EX1002 ¶¶184–185.)

(2) “ACK+ATO”

Regarding the right-hand side of Figure 2, the **initiator** sends an “ACK+ATO” segment to the **responder**. (EX1006, FIG. 2; *see also id.*, 5–7; *supra* Section IX.A.2.c.2.) The **responder**, who “initially proposed the Abort Timeout Option” (*id.*, 7) in the “SYN/ACK+ATO” segment, “analyzes the next segment it receives from its peer” (*id.*) to identify the responsive abort timeout value. The responsive abort timeout value specified in the “ACK+ATO” segment is used by both the **initiator** and **responder** for the connection. (*Id.*) Thus, the **initiator** sends the “ACK+ATO” segment to the **responder** to provide the responsive abort timeout for the abort timeout of the connection. (EX1002 ¶¶186–187.)

* * *

Accordingly, *Eggert* discloses limitation 33.d. (EX1002 ¶188.)

- e) **for use by the another node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.**

Eggert discloses this limitation. (EX1002 ¶¶189–199.) *Eggert* discloses that the abort timeout value (“metadata”) is for use by the **responder** (“another node”) to modify a timeout attribute associated with the connection (“TCP-variant connection”) in at least two ways. (*Id.* ¶¶190–191.)

Figure 2 reveals the two scenarios for ATO negotiation, both of which disclose limitation 33.e.

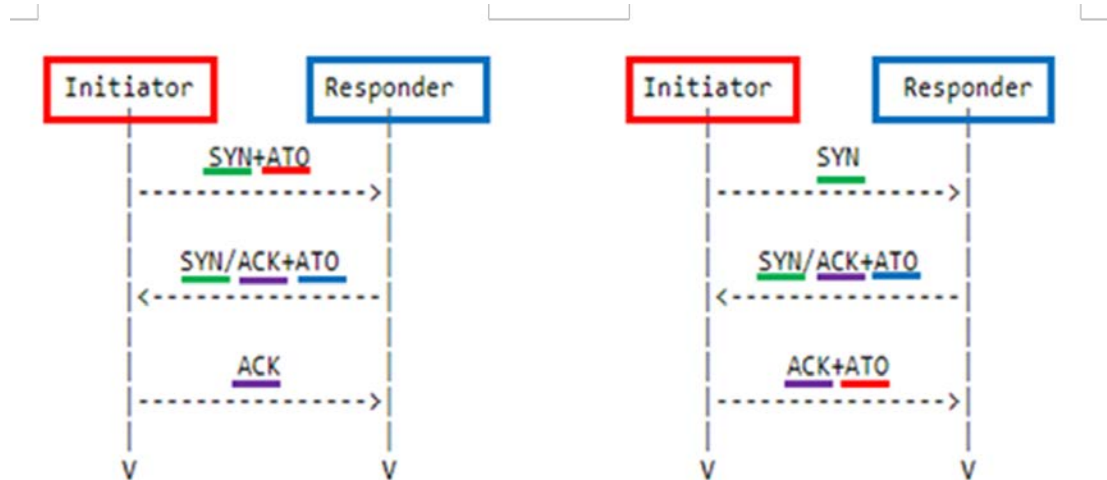


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

(EX1006, FIG. 2 (annotated); EX1002 ¶192.)

(1) “SYN+ATO”

Regarding the left-hand side of Figure 2, because the **initiator** proposed the abort timeout value (“metadata”), the **responder** decides whether to accept, shorten, or reject the proposed abort timeout, resulting in the responsive abort timeout value (“timeout attribute”) for the TCP-variant connection. (EX1006, 5–7, 9, FIG. 2.) For example, “[t]o shorten the offer,” the **responder** uses the proposed timeout value (“metadata”) as an upper bound on the responsive timeout value (“timeout attribute”) (*id.*, 11)—i.e., the **responder** “*lowers the timeout* value accordingly

before sending” (*id.*, 9). (*See also id.*, 5–7.) Thus, the received proposed abort timeout value in the “SYN+ATO” segment is used by the **responder** to modify the responsive abort timeout value associated with the TCP-variant connection. (EX1002 ¶¶193–195.)

(2) “ACK+ATO”

Regarding the right-hand side of Figure 2, because the **responder** initiated an abort timeout negotiation by proposing an abort timeout value, the **responder** must use the responsive abort timeout value contained inside the ATO to modify its abort timeout (“timeout attribute”). (EX1006, 3, 7.) For example, the **responder**, who initially proposed the abort timeout value in the “SYN/ACK+ATO” segment, analyzes the received “ACK+ATO” for the responsive abort timeout value contained inside the ATO to modify its abort timeout associated with the connection to use the same abort timeout as the **initiator** which can shorten the initially proposed timeout value. (*Id.*, 1–3, 7.) Thus, the received responsive abort timeout value (“metadata”) in the “ACK+ATO” segment is used by the **responder** to modify its abort timeout (“timeout attribute”) associated with the TCP-variant connection. (EX1002 ¶196.)

* * *

Petition for *Inter Partes* Review
Patent No. 10,306,026

As discussed above, the '026 patent describes timeout attributes broadly and consistently with *Eggert*'s abort timeout value. (*Supra* Section IX.A.1.e; EX1002 ¶¶197–198.)

Accordingly, *Eggert* discloses limitation 33.e. (EX1002 ¶199.)

B. Ground 2: The Combination of *Eggert* and *Abdolbaghian* Renders Obvious Claims 1, 17, 21–22, 33, 43, 45–48¹¹**1. Claims 1 and 33**

As explained in Ground 1, *Eggert* discloses the “method” of claims 1 and 33. (*Supra* Sections IX.A.1–IX.A.2.) While *Eggert* discloses processes that meet the steps of method claims 1 and 33 (*id.*), to the extent *Eggert* is challenged or is found to not expressly describe a “method” that is performed that includes the processes/protocol described by *Eggert*, a POSITA would have found it obvious to implement *Eggert*’s disclosed processes in a system where host(s) would perform a “method” that followed the protocol/processes described by *Eggert* (discussed above for claims 1 and 33 in Ground 1). This would have been obvious because *Eggert* explains that the disclosed ATO option is for implementation by hosts that wish to negotiate a specific abort timeout for a connection. (EX1006, 3, 5-8.) Therefore, a POSITA would have been motivated to configure a system with hosts communicating over a network to implement and perform a method that includes processes consistent with those described by *Eggert* (and addressed further below) to allow such a system to obtain the benefits provided by *Eggert*’s disclosed process.

¹¹ The grounds pertaining to claims 17, 21–22, 43, and 45–48 apply under both bases for independent claims 1 and 33 presented in Ground 1 and Ground 2.

Petition for *Inter Partes* Review
Patent No. 10,306,026

(*See infra* Sections IX.A.1.a–e, IX.A.2.a–e; EX1002, ¶¶200–201.) Given the disclosures and the knowledge of a POSITA at the time, there would have been a reasonable expectation of success of such an implementation, especially since *Eggert*’s process is designed for use in a system with working computing devices (e.g., hosts) known at the time, and such an implementation would have involved the use of known technologies (e.g., known computing devices and associated software/hardware) that performed a method in accordance with known processes and protocols (e.g., those described by *Eggert*). (EX1002, ¶¶202–203; *see also* EX1001, 23:15–24:13 (describing exemplary generic implementations of the functionality disclosed in the ’026 patent).) Indeed, a POSITA would have been motivated to implement such a “method” in light of the disclosures of *Abdolbaghian*, which discloses a system and method for managing client/server connections in a networked environment similar to *Eggert*. (EX1002 ¶204; *see also* EX1007, Abstract, 1:14–28, 1:49–50, 3:1–6, FIG. 1, cl. 1; *infra* Section IX.B.2.a(2).) Thus, in addition to the reasons above, a POSITA would have been motivated and found obvious to implement *Eggert*’s processes as “method” in view of the disclosures of

Petition for *Inter Partes* Review
Patent No. 10,306,026

Abdolbaghian and the reasons for combining *Eggert* with *Abdolbaghian* discussed below in Sections IX.B.2.a(1)–(2).¹²

2. Claim 17

- a) **The method of claim 1 wherein one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.**

Eggert, in combination with *Abdolbaghian*, discloses and/or suggests this limitation. (EX1002 ¶¶205–231.)

As discussed above, the ATO allows the nodes to negotiate how long each node should keep the connection alive. (*Supra* Sections IX.A.1.a–e.) For example, *Eggert* explains that the ATO “allows mobile hosts to maintain TCP connections across disconnected periods.” (EX1006, 3; *id.*, Abstract.)

While *Eggert* discloses a mechanism to survive a period of disconnection, *Eggert* does not expressly describe other mechanisms—such as a keep-alive mechanism—that are used once the connection has been established to continue to

¹² There is no evidence of nonobviousness at this time that would overcome the overwhelming evidence of obviousness as presented herein. Petitioner reserves the right to address any such evidence/argument should PO later present any in this proceeding.

Petition for *Inter Partes* Review
Patent No. 10,306,026

maintain a connection between the nodes during periods of idleness and disconnection. (EX1002 ¶208.) As discussed below, it would have been obvious to implement the well-known keep-alive mechanism in *Eggert*'s TCP-variant protocol connection (performed in *Eggert*'s "method" as discussed above) such that, e.g., the **initiator**, communicates one or more keep-alive packets, based on a keep-alive period that is, in turn, based on the idle time period to maintain the TCP-variant connection based on the teachings/suggestions of *Abdolbaghian* and the knowledge of a POSITA. (*Id.* ¶¶208–209.)

Indeed, the '026 patent itself acknowledges that the "TCP keep-alive option is supported by a number of implementations of the TCP" (EX1001, 1:54–56) and advantages of the keep-alive option include "detecting a dead peer/partner endpoint sooner" (*id.*, 2:10–11), "detecting when a network is so congested that two nodes with endpoints in a TCP connection are effectively disconnected" (*id.*, 2:12–14), and "keep[ing] an inactive TCP connection open" (*id.*, 2:14–16).

(1) Abdolbaghian

Abdolbaghian, like *Eggert*, relates to managing client/server connections in a networked environment. (EX1002 ¶¶210, 84–91; *see also* EX1007, 1:14–28, 1:49–50, 3:1–6, FIG. 1, cl. 1.) For example, *Abdolbaghian* relates to a communication

Petition for *Inter Partes* Review
 Patent No. 10,306,026

network 4 comprised of connected computers 3a–c and the user terminals 1a–c.
 (EX1007, 2:8–15, FIG. 1.)

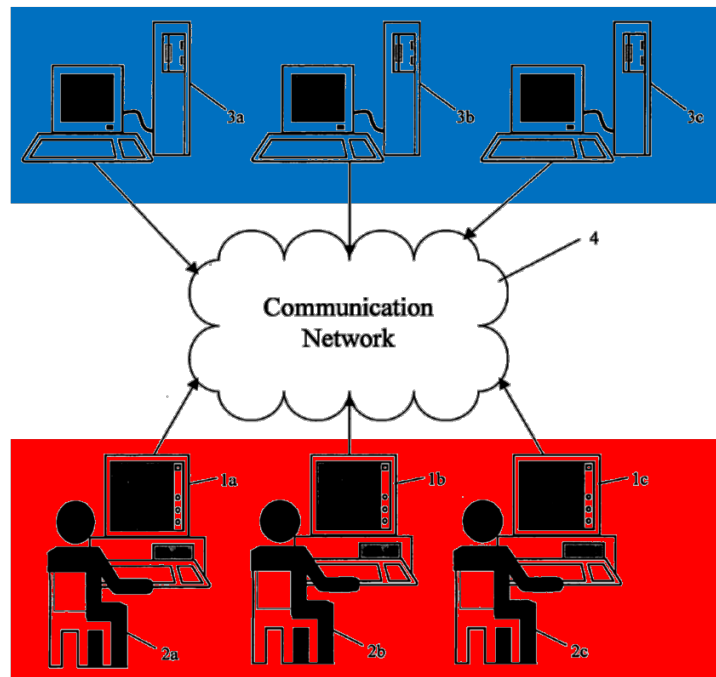


FIGURE 1

(*Id.*, FIG.1; EX1002 ¶210.)

Both *Eggert* and *Abdolbaghian* also disclose sharing timeout information for maintaining a connection. (*Compare* EX1007, 4:28–37 with EX1006, 3–7.) Accordingly, a POSITA would have had reason to consider the teachings of *Abdolbaghian* when implementing *Eggert*'s process, given they disclose features in

Petition for *Inter Partes* Review
Patent No. 10,306,026

a common technical field and address similar problems. (EX1002 ¶¶211-212; *see also infra* Section IX.B.2.a.2.)

Abdolbaghian explains that user terminal 1 and connected computer 3 communicate over a “communication network 4 [which] may include wire line ... or wireless [] connections.” (EX1007, 2:41–44.) Further, *Abdolbaghian* provides that “[i]nformation communicated over the communication network 4 may conform to any data communications protocol, including TCP/IP.” (*Id.*, 2:42–44; EX1002 ¶212.)

Abdolbaghian explains that when the user terminal is operating an application that is, e.g., hosted by the connected computer, it is often undesirable to have that application timeout (e.g., terminated) based on an application timeout period. (EX1007, 1:14–36, EX1002 ¶¶213.) To solve this timeout problem, *Abdolbaghian* provides a keep-alive mechanism that “sends a ping, message or other signal to a first application to prevent the first application from being timed out.” (EX1007, 1:63–67.) The keep-alive message is sent when the keep-alive function timeout clock expires to reset the timeout clock associated with the session in order to maintain the session (e.g., prevent the timeout of the first application). (*Id.*, 4:8–11; EX1002 ¶214.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

Abdolbaghian's "message," "ping," or "other signal," refer to data sent over the communications network 4, and thus are exemplary forms of a packet. (EX1002 ¶215.) Accordingly, *Abdolbaghian* discloses that the user terminal is configured to communicate "one or more keep-alive packets," as claimed. (*Id.*)

Abdolbaghian also discloses that user terminal 1 sets a keep-alive function timeout clock for its keep-alive function based on the longer application timeout period. (EX1007, 1:63–2:7, 4:25–37, FIG. 3.) For instance, "[b]ased on [timeout function] information, the keep-alive function may set its own function timeout clock (the 'process clock')." (*Id.*, 5:21–24; *see also id.*, Abstract.) The "[keep-alive] function timeout clock may be synchronized with the application timeout clock." (*Id.*, 5:24–25.) *Abdolbaghian* also provides that "[t]he value of the [keep-alive] function timeout clock may be set so that the [keep-alive] function timeout clock will expire before the application timeout clock expires." (*Id.*, 5:25–27; *see also id.*, 5:43–46.) Thus, the keep-alive mechanism in *Abdolbaghian* has a clock or timer based on a keep-alive period that is separate from the application's timeout clock or timer, but the keep-alive period is based on the application's timeout period. (EX1002 ¶¶216–221.)

Accordingly, *Abdolbaghian* discloses the concept of setting a keep-alive period that would have kept, e.g., an application on a connection alive, to be shorter

Petition for *Inter Partes* Review
Patent No. 10,306,026

than a longer timeout period based on which the application itself would timeout (e.g., due to disconnection). (*Id.* ¶222.)

(2) Reasons to Combine

A POSITA would have been motivated in light of *Abdolbaghian* to configure the method in *Eggert* (as discussed above for limitations 1.a and 33.a in Grounds 1 and 2 (Sections IX.A.1, IX.B.1)) so that the **initiator** implements the above-discussed keep-alive mechanism to send a keep-alive packet at a frequency set to *expire before* the abort timeout timer expires in the connection. (EX1002 ¶223.) Such an implementation would have been a predictable and straightforward combination of well-known technologies using known methods. *See KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 416–18 (2007).

For example, implementing *Abdolbaghian*’s teachings regarding setting a keep-alive period based on the timeout period (like the abort timeout value in the *Eggert* process) with the method in *Eggert* would have allowed for the combined process to reset the abort timeout timer. (EX1002 ¶¶224.) A POSITA facing the wide range of needs created by developments in the technical field of *Eggert* and *Abdolbaghian* would have appreciated the benefits of sending a keep-alive packet in *Eggert*’s process, based on a keep-alive period that is, in turn, based on the idle time period. (*Id.*)

Petition for *Inter Partes* Review
Patent No. 10,306,026

For instance, given *Eggert*'s process allows hosts to negotiate an abort timeout period for the connection to maintain the established connection across periods of disconnection (EX1006, 3), a POSITA would have appreciated that the benefits to *Eggert* are similar to those provided by *Abdolbaghian*, as discussed above. (EX1002 ¶225; *see also* EX1007, Abstract.) Also, the setting of a keep-alive period in a keep-alive mechanism to be less than a longer timeout value (such as the abort timeout value in the *Eggert* process) for a connection was a well-known concept in the art before the '026 patent. (EX1002 ¶226 (citing EX1012–EX1014).) Thus, setting the keep-alive period based on a longer timeout period would have been a common-sense approach since the purpose of a keep-alive mechanism in the networking field is to keep something (e.g., the connection between two nodes) alive before the connection is subject to be terminated (e.g., based on a longer timeout value such as the abort timeout value). (EX1002 ¶227.)

Therefore, based on *Abdolbaghian*'s disclosure and knowledge in the art, the keep-alive time period in the combined *Eggert–Abdolbaghian* method would have been set to a time period that is *less than* any session's negotiated abort timeout value because to set the time period to a larger (or the same) value as the abort timeout value would render the keep-alive mechanism impracticable. (*Id.* ¶228.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

Furthermore, setting of a keep-alive timer with a period less than the timeout value in the above-combination would have been a straightforward application of basic networking principles for maintaining a connection, as acknowledged by the '026 patent itself. (EX1001, 1:54–46, 2:10–16; EX1002 ¶229 (citing EX1012–EX1015).) Such knowledge, coupled with the disclosures/suggestions of *Abdolbaghian*, would have motivated a POSITA to set the keep-alive timer in the combined process to a value less than the negotiated abort timeout value in order to prevent termination of the connection between the **initiator** and **responder**. (EX1002 ¶230.)

The combination of *Eggert* with *Abdolbaghian* would have involved the use of known technologies (e.g., aspects of similar protocols including the keep-alive mechanism) and design concepts and processes to obtain the foreseeable result of a reliable connection between *Eggert*'s **initiator** and **responder**. (EX1002 ¶¶231–232.) Indeed, the above-modification would have involved implementing a keep-alive mechanism at an **initiator** with a keep-alive timer set less than the negotiated abort timeout value, as discussed above. (*Id.*) A POSITA would have been skilled and knowledgeable to configure *Eggert*'s method to implement a keep-alive mechanism, with a timer set less than the negotiated abort timeout value, while considering any known programming, design, and other related concepts,

Petition for *Inter Partes* Review
Patent No. 10,306,026

limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. (*Id.*) Thus, a POSITA would have had a reasonable expectation of success in the above-modification. (*Id.*) See *Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1364 (Fed. Cir. 2007) (“only a reasonable expectation of success, not a guarantee, is needed” in an obviousness analysis).¹³

Accordingly, the combination of *Eggert* and *Abdolbaghian* discloses and/or suggests claim 17. (EX1002 ¶232.)

¹³ See *supra* n.12.

3. Claim 21

- a) **The method of claim 17 wherein the keep-alive period is administered using a keep-alive timer.**

The *Eggert-Abdolbaghian* combination discloses and/or suggests this limitation for the reasons below and those discussed above for claim 17. (*Supra* Section IX.B.2; EX1002 ¶¶233–235.) As discussed for claim 17, the keep-alive mechanism is configured at an **initiator** in the *Eggert-Abdolbaghian* combination to send one or more keep-alive packets based on a keep-alive function timeout clock (“keep-alive timer”) that dictates the frequency for sending the keep-alive packets, where the keep-alive function timeout clock is based on the abort timeout (“idle time period”), and this discloses and/or suggests this claim. (*Supra* Section IX.B.2; EX1002 ¶¶234–235.)

4. Claim 22

- a) **The method of claim 21 wherein the keep-alive timer is separate from an idle timer that is used to administer the idle time period.**

The *Eggert-Abdolbaghian* combination discloses and/or suggests this limitation for the reasons below and those above for claims 17 and 21. (*Supra* Sections IX.B.2–3; EX1002 ¶¶236–239.) For example, as discussed, the **initiator** in the *Eggert-Abdolbaghian* combination would have included a keep-alive function clock (“keep-alive timer”) that is set based on the abort timeout (“idle time period”).

Petition for *Inter Partes* Review
Patent No. 10,306,026

(*Supra* Sections IX.B.2–3.) Also discussed above, the negotiated abort timeout in *Eggert* is used to detect a time at which the connection between **initiator** and **responder** is subject to deactivation. (*Supra* Sections IX.A.1.d–e.) *Eggert* explains that the abort timeout duration is used to track periods during which there is no communication to keep the connection active. (EX1006, 1–3, 9.) Thus, to detect the expiration of the abort timeout, the combined method would have implemented a timer based on the abort timeout value. Indeed, separate timers could be configured and maintained to track separate time periods in networking applications such as those described in *Eggert* and *Abdolbaghian*. (EX1007, Abstract, 5:15–36 (discussing separate clocks for the keep-alive function and application timeout function); EX1002 ¶238 (citing EX1012, EX1015).) Thus, the *Eggert-Abdolbaghian* combination discloses and/or suggests a method where a keep-alive function clock (“keep-alive timer”) is separate from another timer used to administer the abort timeout (“idle time period”), which is based on the abort timeout value. (EX1002 ¶239.)

5. Claim 43

- a) **The method of claim 33 wherein one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.**

The *Eggert-Abdolbaghian* combination discloses and/or suggests this limitation for at least the same reasons discussed above for limitation 17. (*Supra* Sections IX.A.2, IX.B.2; EX1002 ¶¶240.)

6. Claim 45

- a) **The method of claim 43 wherein the keep-alive period is not configurable, while the idle time period is configurable.¹⁴**

The *Eggert-Abdolbaghian* combination discloses and/or suggests this limitation. (EX1002 ¶¶241–245.) As discussed above, the **initiator** in the *Eggert-Abdolbaghian* combination implements a keep-alive timer set to expire before

¹⁴ The '026 patent specification does not provide any disclosure describing how the “keep-alive period is not configurable” given that the “keep-alive period” is “based on the idle time period” (claim 43) which “is configurable” (claim 45). (EX1002 n.8.) Therefore, while Petitioner addresses such features with respect to the asserted prior art, Petitioner does not concede that the '026 patent provides adequate disclosures of such claimed features.

Petition for *Inter Partes* Review
Patent No. 10,306,026

expiration of the abort timeout timer. (*Supra* Section IX.B.5; *see also supra* Sections IX.B.2–4.)

The abort timeout (“idle time period”) in this combined *Eggert-Abdolbaghian* method “is configurable” because the nodes negotiate the timeout period (e.g., the abort timeout is not set to a default value). (*Supra* Sections IX.A.2.b–e.)

The keep-alive period in the combined *Eggert-Abdolbaghian* method “is not configurable” by the **initiator** at least because *Abdolbaghian* explains that “the keep-alive function may set a function timeout clock to a ***duration known or believed to be shorter*** than the timeout process clock.” (EX1007, 5:43–46.) A known duration for the keep-alive period is one that is fixed (“not configurable”). (EX1002 ¶¶242–243; EX1013, 5:48–63.)

Additionally, given that the keep-alive period in the combined *Eggert-Abdolbaghian* method is set based on the abort timeout, i.e., the keep-alive period is not negotiated during the connection setup process, the keep-alive period is “not configurable” in this additional way. (EX1002 ¶244.) This is consistent with PO’s view of these features. (EX1017, 43 (PO alleging that “[b]ased on the list of negotiable parameters ..., the keep-alive period is not configurable during connection setup,” and explaining that this keep-alive period is “constrained by the idle time period, which is negotiated at connection setup.”).)

Petition for *Inter Partes* Review
Patent No. 10,306,026

Thus, the *Eggert-Abdolbaghian* combination discloses/or suggests claim 45.

(EX1002 ¶245.)

7. Claim 46

- a) **The method of claim 43 wherein the keep-alive period is administered using a keep-alive timer that is separate from an idle timer that is used to administer the idle time period.**

The *Eggert-Abdolbaghian* combination discloses and/or suggests this limitation for at least the same reasons discussed above for claims 21 and 22. (*Supra* Sections IX.A.2, IX.B.3–5; EX1002 ¶246.)

8. Claim 47

- a) **The method of claim 43 wherein the one or more keep-alive packets are conditionally communicated based on at least one keep-alive attribute.**

The *Eggert-Abdolbaghian* combination discloses and/or suggests this limitation in at least two ways. (EX1002 ¶¶247–250.) First, the *Eggert-Abdolbaghian* combination discloses and/or suggests this limitation for at least the same reasons discussed above for limitation 17, where the keep-alive period is a keep-alive attribute because it is information used to determine the property of the keep-alive mechanism in the combination. (*Supra* Sections IX.A.2, IX.B.2, IX.B.5; EX1002 ¶247.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

Second, *Abdolbaghian* further discloses and/or suggests that the sending of one or more keep-alive signals is conditioned on whether the keep-alive mechanism option (“keep-alive attribute”) is enabled or not. (EX1002 ¶¶248–249.) For example, “user 2 may be asked whether the keep-alive process should be initiated and the keep-alive may be initiated only if the user 2 responds affirmatively to this inquiry.” (EX1007, 4:21–24.) Indeed, the optional TCP keep-alive mechanism and its associated parameters was well-known in TCP implementations, as acknowledged by the ’026 patent itself. (EX1001, 1:54–56; EX1002 ¶248 (citing EX1021)). Accordingly, for reasons similar to those discussed above for claim 17, it would have been obvious to incorporate the above features in the *Eggert-Abdolbaghian* combination. (*Supra* Sections IX.B.2.a.2.) For instance, it would have been obvious to further configure the above-discussed *Eggert-Abdolbaghian* combination such that the **initiator** would have conditionally communicated one or more keep-alive packets based on whether the keep-alive option is enabled. (EX1002 ¶249.)

Thus, the *Eggert-Abdolbaghian* combination discloses and/or suggests claim 47. (*Id.* ¶250.)

9. Claim 48

- a) **The method of claim 47 wherein the keep-alive attribute is separate from the timeout attribute.**

The *Eggert-Abdolbaghian* combination discloses and/or suggests the limitations of claim 48. (EX1002 ¶¶251–253.)

First, the *Eggert-Abdolbaghian* combination discloses and/or suggests this limitation for at least the same reasons discussed above for claims 17, 21, and 22. (*Supra* Sections IX.A.2, IX.B.2–4.) For example, the negotiated abort timeout value (“timeout attribute”) is used to specify the keep-alive period (“keep-alive attribute”) such that the keep-alive period is set to a duration shorter than the negotiated abort timeout period. (EX1002 ¶251.)

Second, the *Eggert-Abdolbaghian* combination discloses and/or suggests that the negotiated abort timeout value (“timeout attribute”) is separate from the keep-alive option (“keep-alive attribute”) given that the **initiator** may enable/disable a keep-alive mechanism for the connection. (*Supra* Section IX.B.8.)

Thus, the *Eggert-Abdolbaghian* combination discloses and/or suggests claim 48. (EX1002 ¶¶253.)

C. Ground 3: The Combination of *Eggert* and *Tucker* Render Obvious Claims 31–32 and 51**1. Claim 31**

- a) **The method of claim 1 wherein a duration of the idle time period is configured by an application via an analog of a socket.**

Eggert in view of *Tucker* discloses and/or suggests the limitations of claim 31. (EX1002 ¶¶254–274.) As discussed above for claim 1, *Eggert*’s **initiator** initiates the process for establishing a TCP-variant connection with **responder** that includes the negotiation of an abort timeout value (“a duration of the idle time period”). (*Supra* Sections IX.A.1.a–g.)

For example, *Eggert* discloses that the **initiator** via its application performs the abort timeout value negotiation with another node during the connection’s three-way handshake. (EX1006, 9; *see also id.*, 5.) A POSITA would have thus recognized that the abort timeout for the connection in *Eggert* is configured—e.g., negotiated—by an **initiator**’s application via an interface between the application and the TCP-variant protocol. (EX1002 ¶¶255–256.)

An inter-process communication (“IPC”) would be used to provide the interface between the application/user and the TCP-variant protocol. (*Id.* ¶257; *see also infra* Section IX.C.1.a.1 (explaining that a socket is one example of an IPC implementation).) As discussed, *Eggert*’s TCP-variant protocol is a variation of the

Petition for *Inter Partes* Review
Patent No. 10,306,026

TCP protocol described in RFC 793 as it includes functionality for negotiating an abort timeout during the three-way handshake. (*Supra* Sections IX.A.1.a–b.) According to RFC 793, the TCP protocol includes an IPC involving the interface between the protocol and the application/user. (EX1008, 49 (“[i]n providing interprocess communication facilities, the TCP must not only accept commands, but must also return information to the processes it serves”); *see also id.*, 7.) Accordingly, *Eggert’s* TCP-variant protocol would have similarly included an interface to allow the exchange of data between the protocol and the application/user—for example, an interface for facilitating data exchange related to an abort timeout value (EX1006, 9). (EX1002 ¶257.)

This interface would have allowed calls for exchanging data between the user programs (e.g., application) and the TCP-variant protocol. (*Id.* ¶258; EX1008, 13 (“[t]he TCP/user interface provides for calls made by the user on the TCP interface.”); *see also id.*, 49.) “These calls are like other calls from user programs on the operating system, for example, the calls to open, read from, and close a file” (EX1008, 9) and thus “specify the basic functions the TCP must perform to support interprocess communication” (*id.*, 48). (Ex. 1002 ¶258.)

While *Eggert’s* TCP-variant protocol, in light of RFC 793, provides an IPC between the application and the TCP-variant protocol (e.g., at the **initiator**) to

Petition for *Inter Partes* Review
Patent No. 10,306,026

facilitate the configuration of an abort timeout value for negotiation with the **responder**, *Eggert* does not expressly provide the specific IPC mechanism used to implement the interface. Nevertheless, it would have been obvious to implement *Eggert*'s interface between the protocol and the application/user with an “analog of a socket” to allow the exchange of a timeout value between the application and protocol based on the teachings/suggestions of *Tucker* and a POSITA's knowledge. (EX1002 ¶259.)

(1) Tucker

Tucker, like *Eggert*, relates to the communication of messages and data between different processes. (EX1002 ¶¶260, 92–93; *compare* EX1036, Abstract with EX1006, 9.) *Tucker* introduces common IPCs typically found in an operating system (OS), such as “a shared memory segment, a message queue, . . . , a pipe, a stream, a socket” (EX1036, 3:36–43), which are utilized by an operating system to allow communication (e.g., the exchange messages or commands) between different processes or programs. (*See also* EX1036, 2:36–39 (explaining that an operating system includes IPCs), 3:11–15 (explaining that authorized processes may use IPC components for communication), 3:45–50, 8:52–56, 9:37–45 (disclosing the various IPC implementations), 10:24–31 (explaining that there are a variety of mechanisms available for providing IPC capability), FIG. 2A; EX1002 ¶261.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

A POSITA would have understood that a designer may choose from a variety of IPC objects when enabling IPC capability between processes (e.g., a user application and a TCP-type protocol as discussed above) to meet different system objectives, such as performance, modularity, and system circumstances. (*Id.*, ¶262; EX1036, 10:24–26.) Accordingly, to enable communications between processes, a designer may select an available IPC mechanism in an OS environment. (EX1036, 8:59–61, 10:54–63 (describing the creation and use of a communication object), FIG. 2A; EX1002 ¶262.)

As with any IPC implementation, a socket allows data communication to be sent over a network interface, either to a different process on the same machine or to another machine on the network. (EX1002 ¶263; EX1036, 13:16–22, 14:50–56.) *Tucker* explains that a socket can be interchanged with other IPC components such as a shared memory segment, a message queue, a pipe, or a stream for establishing an IPC. (EX1036, 13:55–65.) These non-socket IPC components are similar to a socket component in that they each facilitate communication between processes on a machine. (EX1002 ¶263.) Thus, a shared memory segment, a message queue, a pipe, and a stream are each an analog of a socket component for providing inter-process communication. (EX1002 ¶264; *see also id.* ¶¶263–264 (discussing other IPC components in *Tucker* that are also analogs of a socket).)

Petition for *Inter Partes* Review
Patent No. 10,306,026

Tucker's disclosures regarding analogs of a socket are consistent with the teachings of the '026 patent. For example, the '026 patent explains that the socket—and the analog of a socket component—allows data exchange between the network application layer and the TCP layer, which is illustrated in Figure 5. (*E.g.*, EX1001, 11:2–4, 12:11–13, 12:23–25.) While the '026 patent does not provide an explicit definition of what an analog of a socket is, the '026 patent discloses other communication mechanisms within a system, including “a message queue,” “pipe,” or “shared location in IPU memory and/or secondary storage” like *Tucker*'s above-noted message queue, pipe, and shared memory segment. (*Id.*, 11:65–12:3; EX1002 ¶265.)

Moreover, the understanding that *Tucker*'s descriptions regarding a stream teaching an analog of a socket is consistent with PO's view of the same claimed features. (EX1017, 20–22 (mapping an “analog of a socket” to a “stream socket”).)

(2) Reasons to Combine

Based on the teachings of *Tucker* and a POSITA's knowledge at the time, it would have been obvious to configure *Eggert*'s method (as discussed for claim 1) to implement the interface between the TCP-variant protocol and the application/user at *Eggert*'s node (e.g., **initiator**) to use non-socket IPC components, such as a shared memory segment, a message queue, a pipe, or a stream (each disclosing “an analog

Petition for *Inter Partes* Review
Patent No. 10,306,026

of a socket”) to enable the user application to configure the abort timeout value (“the duration of the idle time period”) for the TCP-variant connection between the user application and the TCP-variant protocol. (EX1002 ¶266.) Such an implementation would have had many advantages and been a predictable and straightforward result achieved by combining well-known technologies using known methods. *See KSR*, 550 U.S. at 416–18.

Tucker and *Eggert* disclose features in a similar technological field. (Ex. 1002 ¶267; *supra* Section IX.C.1.a.1.) Like *Eggert*, *Tucker* discloses features for allowing the communication of data and commands between different processes (e.g., EX1036, 10:24–31), so a POSITA would have had reason to consider *Tucker* when contemplating and implementing the teachings of *Eggert*.

A POSITA would have been motivated to incorporate *Tucker*’s teachings and suggestions of creating and using an IPC component, such as the above-discussed non-socket IPC components, for *Eggert*’s interface (e.g., at the **initiator**) to facilitate the configuration of an abort timeout value between the user application and the TCP-variant protocol for negotiation of an abort timeout with the **responder**. (*Id.* ¶268.) Using any one of these non-socket IPC components as described in *Tucker* in *Eggert*’s system/process as the mechanism to facilitate exchange of an abort timeout between the user application and the TCP-variant protocol would have

Petition for *Inter Partes* Review
Patent No. 10,306,026

achieved the required communication between the user application and the TCP-variant protocol (e.g., at the **initiator**). (*Id.* ¶268.) Indeed, IPC implementations, like those disclosed in *Tucker*, are applicable in wide variety of networking environments including both TCP and UDP based environments. (*Id.* ¶268 (citing EX1038–EX1039).) Furthermore, the above-modification would have involved the substitution of one known IPC component (e.g., a socket) with another known non-socket IPC component from a finite number of available IPC components such as those described by *Tucker*. (*Id.* ¶269.)

Given that the **initiator**'s application in *Eggert* must communicate with the TCP-variant protocol as discussed above, configuring *Eggert* to utilize a non-socket IPC component for the intercommunication between the user application and the TCP-variant protocol based on the teachings of *Tucker* would have been both a predictable and straightforward implementation. (*Id.* ¶270); see *KSR*, 550 U.S. at 416. A POSITA would have had the skills and knowledge to achieve this implementation by combining well-known technologies using known methods (such as the above-discussed non-socket IPC components, which are common mechanisms in operating systems), and technologies as described above by *Tucker* and *Eggert*, and known in the art at the time. (EX1002 ¶270.) Accordingly, such a

Petition for *Inter Partes* Review
Patent No. 10,306,026

combination would have yielded the predictable result of providing an IPC between the application and the TCP-variant protocol. (*Id.* ¶271); *see KSR*, 550 U.S. at 417.

Also, the *Eggert-Tucker* combination would have involved the use of known technologies (e.g., IPC components) and design concepts and processes to obtain the foreseeable result at a node of an IPC between the application and the TCP-variant protocol using the above-discussed non-socket IPC components. (EX1002 ¶272.) Thus, a POSITA would have known how to configure *Eggert*'s method to implement *Tucker*'s above-discussed non-socket IPC component implementations, and would have had a reasonable expectation of success in the above-modification. (*Id.* ¶273.) *Pfizer*, 480 F.3d at 1364.

* * *

Thus, the *Eggert-Tucker* combination discloses and/or suggests claim 31. (EX1002 ¶274.)

2. Claim 32

- a) **The method of claim 1 wherein the metadata is identified by an application via an analog of a socket.**

The *Eggert-Tucker* combination discloses and/or suggests this limitation for similar reasons as discussed above for claim 31. (*See* Section IX.C.1; *see also* EX1002 ¶¶275–277.) For example, *Eggert* discloses that the **initiator**, by its application, performs abort timeout value negotiation with the **responder**.

Petition for *Inter Partes* Review
Patent No. 10,306,026

(EX1006, 9 (“applications requesting such short timeouts”), 5.) Accordingly, the application at the **initiator** identifies the received abort timeout value (“metadata”) from e.g., a “SYN/ACK+ATO” segment, which has been passed to the application from the TCP-variant protocol via an IPC, for establishing the connection. (EX1002 ¶275; *see also supra* Sections IX.A.1.d–e (describing the **initiator**’s identification of “metadata” and its use).)

As explained above, non-socket IPC components would have been used by the application to identify and communicate the abort timeout value to the TCP-variant protocol. (*Supra* Section IX.C.1.) Accordingly, for reasons similar to those explained above for claim 31, a POSITA would have been motivated to further configure the combined *Eggert-Tucker* process such that it includes the process of identifying the metadata at the **initiator** by its application via the above-discussed non-socket IPC components in order to perform the abort timeout value negotiation with the **responder**. (*Supra* Section IX.C.1; EX1002 ¶276.)

* * *

Thus, the *Eggert-Tucker* combination discloses and/or suggests claim 32. (EX1002 ¶277.)

3. Claim 51

- a) **The method of claim 33 wherein a duration of the idle time period is configured by an application via an analog of a socket.**

The *Eggert-Tucker* combination discloses and/or suggests the limitations of claim 51 for the same reasons discussed above for claims 31 and 33. (*See supra* Sections IX.A.2, IX.C.1; EX1002 ¶278.)

D. Ground 4: The Combination of *Eggert*, *Abdolbaghian* and *Tucker* Render Obvious Claims 31–32 and 51

1. Claims 31–32, and 51

As discussed above, the *Eggert-Tucker* combination discloses or suggests the limitations of claims 31–32 and 51. (*Supra* Section IX.C.) *Eggert* in combination with *Abdolbaghian* and *Tucker* discloses or suggests the limitations of claims 31–32 and 51 for at least the same reasons as discussed above in Ground 3, given *Abdolbaghian* provides a “method” in the *Eggert-Abdolbaghian* combination and would not affect the proposed incorporation of *Tucker*’s teachings in *Eggert*’s implementations. (EX1002 ¶279; *supra* Sections IX.B.1, IX.C.)

X. EGGERT IS A PRINTED PUBLICATION

Eggert is an IETF Internet-Draft (“ID”) and is prior art under at least 35 U.S.C. §102(b) (pre-AIA) because it was published in April, 2004. The declaration of Alexa Morris, Managing Director of IETF, confirms *Eggert* was published,

Petition for *Inter Partes* Review
Patent No. 10,306,026

disseminated, and reasonably available to the public by April, 2004. (EX1019 ¶¶1-3, 9–10.)

IDs were at the time, and continue to be, working documents published through the IETF Secretariat and disseminated to the public by IETF through various media so that others may comment on them. (*Id.* ¶¶4–6.) Since 1998 (including 2004 and today), the IETF Secretariat publishes an ID on its public website, and publication announcements were sent to an IETF mailing list and relevant working group mailing lists. (*Id.* ¶¶5–6.) Anyone could have subscribed to IETF mailing lists, and the archives of all IETF mailing lists are publicly available on IETF’s website. (*Id.* ¶¶6–8.)

Eggert was published by the IETF in April, 2004. (*Id.* ¶¶9–10.) In 2004, a POSITA could have learned about *Eggert* in various ways, such as through announcements on the IETF announce mailing list, discussions on the IETF tcpm working group mailing list, or review of the archives of the IETF announce or tcpm mailing lists. (*Id.*)

Collectively, this demonstrates that *Eggert* was published and publicly available in 2004 and at least prior to 2010. *Polycom, Inc., v. Directpacket Research, Inc.*, IPR2019-01235, Paper 19, 30–32 (Jan. 13, 2020) (finding a similar IETF ID was publicly available based on similar evidence).

XI. DISCRETIONARY DENIAL IS NOT APPROPRIATE**A. *Eggert* is Not Cumulative**

Eggert is not cumulative to any prior art considered during prosecution of the '026 patent and no arguments similar to those contained herein were ever presented to or considered by the Office. Nonetheless, PO may assert here (as it did in the Google -845 IPR) that RFC 5482 (EX1016) listed on the face of the '026 patent is substantially similar to *Eggert*. It is not, as the Board found in the Google -845 IPR. Google -845 IPR, Paper 16 at 14–17. The Board should reach the same conclusion here.

First, the Board routinely institutes trial where references in an IPR were considered in an IDS but not relied upon to reject claims during prosecution. *See, e.g., Amber.IO, Inc. D/B/A Two Tap v. 72Lux, Inc. D/B/A Shoppable*, IPR2020-00015, Paper 8 at 18-20 (April 1, 2020) (declining to exercise discretion where references were cited in an IDS but no evidence that the references were substantively considered) (citing *Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6 (February 13, 2020) (precedential)); *Apple, Inc. v. Omni Medsci, Inc.*, IPR2020-00029, Paper 7 at 52–55 (April 22, 2020) (similar). Consistent with those proceedings, RFC 5482 was cited during prosecution along with twenty other references in an IDS with no explanation

Petition for *Inter Partes* Review
Patent No. 10,306,026

regarding any relevant disclosures. (EX1004, 182–184.) Moreover, RFC 5482 was never relied upon to reject the claims. (*Id.*, 151–153) Even assuming *Eggert* is cumulative to RFC 5482, which it is not as explained below, *Eggert* was not considered in the light being presented herein (e.g., with supporting expert testimony).

Second, as the Board previously found, *Eggert*'s protocol is fundamentally different from that discussed in RFC 5482 because, in contrast with RFC 5482, *Eggert*'s protocol calls for a common negotiated value for the timeout. Google -845 IPR, Paper 16 at 14–17. While *Eggert* discloses a negotiation protocol, RFC 5482 was purposefully designed such that “an exchange of UTO [user timeout] options between both ends of a connection is not a binding negotiation.” (EX1016, 4.)

Eggert also has meaningful disclosures that do not appear in RFC 5482. For example, it is *Eggert*'s mechanisms disclosed with reference to Figure 2 regarding exchange of ATO information during the three-way handshake that is relied upon to teach claim limitations 1.b–e and 33.b–e. (*Supra* Sections IX.A.1.b–e, IX.A.2.b–e.) In contrast, RFC 5482 “does not define a mechanism to negotiate support of the TCP User Timeout Option [UTO] during the three-way handshake.” (EX1016, 9.) Therefore, these “material differences between the asserted art and the prior art involved during examination” weigh against discretionary denial. *Becton*,

Petition for *Inter Partes* Review
Patent No. 10,306,026

Dickinson & Co. v. B. Braun Melsungen AG, IPR2017-01586, Paper 8, 17-18 (Dec. 15, 2017) (precedential as to §III.C.5, first paragraph); *see also* Google -845 IPR, Paper 16 at 16–17. Thus, *Eggert* is not cumulative to RFC 5482.

B. The Related Litigation Provides No Basis For Discretionary Denial

The Board should not exercise its discretion to deny institution under 35 U.S.C. § 314(a). *NHK Spring Co., Ltd. v. Intrix-Plex Techs, Inc.*, IPR2018-00752, Paper 8 (Sept. 12, 2018) does not apply here as no trial date has been scheduled in the related litigation. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 3 (Mar. 20, 2020) (precedential) (“*NHK* applies ... where the district court has set a trial date to occur earlier than the Board’s deadline to issue a final written decision in an instituted proceeding.”). The six-factor test addressed in *Fintiv* (“*Fintiv* factor(s)”) favors institution. *See id.*, 5–6.

The **first *Fintiv* factor** (stay) is neutral, if not weighing in favor of institution. While there is currently no stay pending IPR resolution before the Western District of Texas (“WDTX”) court, Google intends to pursue such a stay should institution be granted. The Board should not speculate as to how the court would rule on the requested stay based on actions taken in different cases. *Western Digital Corp. et al. v. Martin Kuster*, IPR2020-01391, Paper 10 at 8–9 (February 16, 2020) (finding this factor neutral despite Patent Owner assertions regarding WDTX Judge

Petition for *Inter Partes* Review
Patent No. 10,306,026

Albright’s past practices regarding stays). Moreover, Google filed a motion to transfer in the district court litigation. The Court has currently stayed the litigation pending resolution of Google’s transfer motion. (EX1035.)

The **second *Fintiv* factor** (proximity of trial dates) weighs strongly in favor of institution. *First*, the district court has not set a trial date, which “weighs significantly against exercising [] discretion to deny institution of the proceeding.” *Google LLC v. Uniloc 2017 LLC*, IPR2020-00441, Paper 13 at 35 (July 17, 2020); *Nested Bean, Inc. v. Big Beings USA PTY LTD*, IPR2020-01234, Paper 15 at 14 (Jan. 25, 2021) (same). In fact, while the parties submitted proposed schedules, the court granted Google’s motion to stay the case pending resolution of Google’s transfer motion and declined to set a case schedule.¹⁵ (EX1035; EX1029.) Accordingly, determining at this time whether the Board’s anticipated FWD date (e.g., September–October 2022) will occur after trial in the related litigation would be speculative.

¹⁵ Disposition of Google’s transfer motion has taken priority over other activities. (EX1035.) *See also In re Apple Inc.*, 979 F.3d 1332, 1337 (Fed. Cir. 2020) (“disposing of a filed transfer motion ‘should unquestionably take top priority.’”).

Petition for *Inter Partes* Review
Patent No. 10,306,026

Even if the parties' most recent proposed schedule was considered as a baseline for comparison (which Google submits would be improper at this time), dates in that proposed schedule would need to be delayed because they were proposed prior to the court's recent stay order and did not take the stay into consideration. (EX1035; EX1029, 5–9.) Accordingly, the proposed dates in light of the court's stay order demonstrates that trial will likely occur after August 2022.¹⁶ (EX1029, 6, 9.) Thus, any projected trial date (*albeit* speculative) would be more likely after the expected due date of the Board's FWD (e.g., around September–October 2022).

The Board has instituted trial even where an *actual* litigation trial date precedes the FWD deadline by two months or more. *Sand Revolution II, LLC v. Cont'l Intermodal Grp–Trucking LLC*, IPR2019-01393, Paper 24 at 8–10 (June 16, 2020) (“*Sand*”) (informative) (factor 2 favored institution despite WDTX trial

¹⁶ Consistent with the court's practice, the proposed order stages deadlines and trial date based on the proposed August 2021 *Markman* hearing and that the “actual trial date” may materially differ from the schedule and that the court would consider “reasonable amendments to the case schedule post-*Markman*.” (EX1029, 9 n.3; *id.*, 6.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

preceding FWD deadline by five months); *Western Digital Corp.*, IPR2020-01391 at 9-10; *Apple Inc. v. Parus Holdings, Inc.*, IPR2020-00686, Paper 9 at 11–13, 22; *SMIC, Am. v. Innovation Foundry Techs., LLC*, IPR2020-00786, Paper 10, at 20–21 (Oct. 5, 2020); *Fintiv*, IPR2020-00019, Paper 11 at 5, 9.

It is also unlikely that trial in WDTX will proceed without delay. WDTX has issued a suspension order every month for the past twelve months suspending almost all trials in the district from March 13, 2020 to at least April 30, 2021 due to the COVID-19 pandemic, creating a large backlog of trials. (*See* EX1030; *see also* EX1031, 2 (agreeing to stipulation to postpone trial date “to allow the COVID-19 situation to ameliorate”); EX1034 (showing J. Albright’s docket including 792 cases filed in 2020).) Thus, as the Board has recognized, civil trials in WDTX “may possibly slip ... [due to] months of backlogged trials, including many active criminal cases that would take precedence over civil trials.” *HP Inc. v. Slingshot Printing LLC*, IPR2020-01085, Paper 12 at 7 (Jan. 14, 2021); *Sand* at 8-10.

Finally, the potential for transfer based on Google’s transfer motion further reduces the likelihood that trial will precede the Board’s FWD. *Dish Network, L.L.C. v. Broadband iTV, Inc.*, IPR2020-01267, Paper 15 at 17–18 (Jan. 21, 2021) (explaining that the second *Fintiv* factor “is a proxy for the *likelihood* that the trial court will reach a decision on validity issues before the Board reaches a [FWD]”).

Petition for *Inter Partes* Review
Patent No. 10,306,026

Indeed, transfer of the WDTX case to the proper venue (the Northern District of California) will undoubtedly delay any trial beyond the anticipated FWD due date. *C.f. Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 8, at 7 n.7 (Mar. 27, 2020); *see Uniloc 2017 LLC v. Google LLC*, Case No. 18-CV-00502, Dkt No. 277 (E.D. Tex. June 19, 2020) (transfer order vacating unreach deadlines).

The **third *Fintiv* factor** (investment in parallel proceedings) also weighs in favor of institution. The case is in earliest stages, and as discussed, all deadlines other than those concerning Google’s transfer motion have been stayed. Thus, investment by the parties in invalidity and claim constructions issues will be delayed, especially if the case is transferred. And even if the transfer motion is denied, using the above-discussed proposed schedule as an exemplary baseline (which will require adjustments), any *Markman* hearing in the case is likely not to occur until August 2021 or later, and thus would occur near the expected due date of the Board’s institution decision. (EX1029, 6.) Discovery will not commence, and dispositive motions will not be due, until after the *Markman* hearing. Other case activity that has taken place or will eventually take place (outside claim construction) that does not relate to validity should “not weigh in [the Board’s] consideration of this issue.” *Western Digital Corp.*, IPR2020-01391 at 11. Such minimal investment is insufficient to support discretionary denial. *Juniper Networks, Inc. v. Huawei*

Petition for *Inter Partes* Review
Patent No. 10,306,026

Digital Techs. (Cheng Du) Co., Ltd., IPR2020-01130, Paper 13 at 12–13 (Jan. 22, 2021) (factor 3 weighs against discretionary denial when claim construction order has been entered, but fact and expert discovery has not closed, and dispositive motions and briefings have not been submitted); *Dish* at 19–21 (similar); *HP* at 7 (similar). Additionally, Google’s diligence in filing this Petition just four months after receiving PO’s narrowed list of asserted claims¹⁷ further weighs against discretionary denial. *Dish* at 20–21; *Fintiv* at 11.

The **fourth *Fintiv* factor** (overlap) is neutral if not in favor of institution. There is, at this early stage of the district court litigation, no evidence of overlap with the grounds here and any invalidity positions to be pursued in the district court. Indeed, preliminary invalidity contentions have not yet been served. And, as

¹⁷ PO initially asserted 455 claims across eight patents (including 107 claims from the ’026 patent). (EX1037, 1.) The WDTX court ordered PO to substantially reduce the asserted claims to approximately 65 claims. On October 20, 2020, PO narrowed its asserted claims as ordered, but “reserve[d] the right to either narrow the claims further, **substitute claims**, or **add a reasonable number of claims**.” (EX1032.) On December 4, 2020, PO informed Petitioner that it will not modify its narrowed list of asserted claims “**at this time**.” (EX1033.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

discussed above, the district court must first resolve the pending transfer motion before addressing substantive issues, including invalidity. Given the stay of the litigation, it is likely expert discovery relating to invalidity will not be completed until at the earliest mid-2022 and dispositive motions on those issues would be due after that. Accordingly, substantive invalidity issues would not be addressed in the litigation until well past the institution decision and likely very near the time of the Board's expected FWD. Moreover, Petitioner may rely on invalidity grounds not asserted here in the district court litigation, including system art and prior art references and combinations not raised herein. Therefore, any potential overlap between this proceeding and the district court is minimal and speculative at this time.

Other circumstances (*Fintiv* factor six) also favor institution. The Petition presents strong grounds demonstrating the unpatentability of the challenged claims, weighing against discretionary denial. (*See supra* Section IX.) *Western Digital Corp.*, IPR2020-01391 at 14-15. Indeed, the Board has already instituted the Google -845 IPR and the Unified -742 IPR challenging claims of the related '995 patent. (*See supra* Section II.) The Google -845 IPR is based on *Eggert*, which is being applied here. Moreover, this petition and the concurrently-filed sister petition (*id.*) are the only challenges to the '026 patent before the Board, which is a "crucial fact"

Petition for *Inter Partes* Review
Patent No. 10,306,026

favoring institution. *Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 10 at 6 (May 12, 2020).¹⁸

While ***Fintiv* factor 5** (same parties) may weigh slightly in favor of denial, the remaining factors are at least neutral, if not strongly favoring institution. Further, even if the Board finds that **factor 1** (or even **factor 2**) somehow tilts in favor of denial, Petitioner’s diligence, the lack of relevant investment of resources, lack of evidence of overlap, and/or strength of Petitioner’s grounds (**factors 3, 4, 6**) outweigh these other factors. *See e.g., SK Hynix Inc. et al. v. Netlist, Inc.*, IPR2020-01421, Paper 10 at 6–13 (Mar. 16, 2021). Accordingly, based on a “holistic view” of whether integrity of the system and efficiency is best served, institution here is proper. *Samsung Elecs. Co. Ltd. v. Dynamics Inc.*, IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

¹⁸ Discretionary denial under *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper No. 19 is not applicable here given the ’026 patent is not at issue in any other proceeding pending before the Board (outside of Petitioner’s concurrently-filed sister petition).

Petition for *Inter Partes* Review
Patent No. 10,306,026

XII. CONCLUSION

For the reasons given above, Petitioner requests institution of IPR for claims 1, 17, 21, 22, 31–33, 43, 45–48, and 51 of the '026 patent based on each of the grounds specified in this petition.

Respectfully submitted,

Dated: April 29, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,306,026

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,306,026 contains, as measured by the word-processing system used to prepare this paper, 13,989 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: April 29, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,306,026

CERTIFICATE OF SERVICE

I hereby certify that on April 29, 2021, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,306,026 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on PAIR:

Patrick Caldwell (Reg. No. 44580)
G. Gordon (Reg. No. 64517)
P.O. Box 59655
Dept. SVIPGP
Dallas, TX 75229

A courtesy copy was also sent via electronic mail to the Patent Owner's litigation counsel at the following addresses:

TD-PTAB@devlinlawfirm.com
ddahlgren@devlinlawfirm.com
dlflitparas@devlinlawfirm.com
1526 Gilpin Ave.
Wilmington, DE 19806

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)

EXHIBIT 10

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

JENAM TECH, LLC,
Patent Owner

Patent No. 10,306,026

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 10,306,026**

Petition for *Inter Partes* Review
Patent No. 10,306,026

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	1
III.	PAYMENT OF FEES	3
IV.	GROUND FOR STANDING.....	3
V.	PRECISE RELIEF REQUESTED	3
A.	Claims for Which Review Is Requested	3
B.	Statutory Ground of Challenge	3
VI.	LEVEL OF ORDINARY SKILL	5
VII.	OVERVIEW OF THE '026 PATENT.....	6
VIII.	CLAIM CONSTRUCTION	9
IX.	DETAILED EXPLANATION OF GROUND.....	12
A.	Ground 1: The Combination of <i>Wookey</i> and <i>Berg</i> Render Obvious Claims 52, 55–57, 59, 61, 64, 66, 68–69.....	12
1.	Claim 52.....	12
2.	Claim 55.....	45
3.	Claim 56.....	47
4.	Claim 57.....	49
5.	Claim 59.....	50
6.	Claim 61.....	50
7.	Claim 64.....	58
8.	Claim 66.....	59
9.	Claim 68.....	59

Petition for *Inter Partes* Review
Patent No. 10,306,026

10.	Claim 69.....	65
B.	Ground 2: The Combination of <i>Wookey</i> , <i>Berg</i> , and <i>Tucker</i> Render Obvious Claims 60 and 67.....	66
1.	Claim 60.....	66
2.	Claim 67.....	74
X.	DISCRETIONARY DENIAL IS NOT APPROPRIATE	75
XI.	CONCLUSION.....	83

Petition for *Inter Partes* Review
Patent No. 10,306,026

LIST OF EXHIBITS

EX1001	U.S. Patent No. 10,306,026
EX1002	Declaration of Bill Lin, Ph.D.
EX1003	Curriculum Vitae of Bill Lin, Ph.D.
EX1004	Prosecution History of U.S. Patent No. 10,306,026
EX1005	U.S. Pre-Grant Publication No. 2007/0171921 to Wookey <i>et al.</i>
EX1006	RESERVED
EX1007	U.S. Patent 6,674,713 to Berg et al.
EX1008	DARPA RFC 793 TRANSMISSION CONTROL PROTOCOL
EX1009	U.S. Patent 6,584,546 to Kavipurapu
EX1010	U.S. Patent No. 9,923,995
EX1011	Bova et al., RELIABLE UDP PROTOCOL <draft-ietf-sigtran-reliable-udp-00.txt> 25 February 1999
EX1012	U.S. Pre-Grant Publication No. 2004/0093376 to De Boor et al.
EX1013	U.S. Patent 7,535,913 to Minami et al.
EX1014	U.S. Pre-Grant Publication No. 2005/0204013 to Raghunath <i>et al.</i>
EX1015	U.S. Pre-Grant Publication No. 2007/0005804 to Rideout
EX1016	U.S. Pre-Grant Publication No. 2004/0098748 to Bo et al.
EX1017	<i>Jenam Tech, LLC's Second Amended Set of Infringement Contentions</i> regarding U.S. Patent No. 10,306,026 (March 17, 2021)

Petition for *Inter Partes* Review
Patent No. 10,306,026

EX10181	IETF RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1
EX1019	U.S. Patent No. 7,636,805 to Rosenberg
EX1020	U.S. Patent No. 6,212,175 to Harsch
EX1021	U.S. Patent No. 8,259,716 to Diab
EX1022	U.S. Patent No. 6,665,727 to Hayden
EX1023	U.S. Patent No. 6,981,048 to Abdolbaghian <i>et al.</i>
EX1024	<i>Jenam Tech., LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Joint Submission of Proposed Amended Scheduling Order, ECF No. 57 (W.D. Tex. Mar. 4, 2021)
EX1025	Fourteenth Supplemental Order Regarding Court Operations Under the Exigent Circumstances Created By the COVID-19 Pandemic, The United States District Court for the Western District of Texas (March 17, 2021)
EX1026	<i>Digital Retail Apps, Inc. v. H-E-B, LP</i> , Case No. 6:19-cv-00167, Joint Stipulation and Order Postponing Trial, ECF No. 182 (W.D. Tex. Jan. 13, 2021)
EX1027	<i>Jenam Tech, LLC's</i> Preliminary Narrowing of Claims (October 20, 2020)
EX1028	<i>Jenam Tech, LLC's</i> Correspondence Regarding Narrowing of Claims (December 4, 2020)
EX1029	Judge Alan D. Albright's Case Statistics By Year (Retrieved from DocketNavigator on March 9, 2021)
EX1030	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Order Granting Motion to Stay Case, ECF No. 58 (W.D. Tex. Mar. 10, 2021)
EX1031	U.S. Patent No. 7,389,512 to Tucker (" <i>Tucker</i> ")
EX1032	<i>Jenam Tech, LLC's First Set of Infringement Contentions</i> regarding U.S. Patent No. 10,306,026 (August 21, 2020)

Petition for *Inter Partes* Review
Patent No. 10,306,026

EX1033	U.S. Pre-Grant Publication No. 2008/0144603 to Chouksey <i>et al.</i>
EX1034	U.S. Pre-Grant Publication No. 2003/0131135 to Yun

Petition for *Inter Partes* Review
Patent No. 10,306,026

I. INTRODUCTION

Google LLC (“Petitioner”) requests *inter partes* review (“IPR”) of claims 52, 55–57, 59–61, 64, and 66–69 (“the challenged claims”) of U.S. Patent No. 10,306,026 (“the ’026 patent”) (EX1001) assigned to Jenam Tech, LLC (“Patent Owner” or “PO”). For the reasons below, the challenged claims should be found unpatentable and canceled.

II. MANDATORY NOTICES

Real Parties-in-Interest: Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies the following as the real parties-in-interest: Google LLC.¹

Related Matters: The ’026 patent is asserted in the following civil action: *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.). The ’026 patent was previously asserted in the following civil action: *Jenam Tech, LLC v. Samsung Electronics Co., Ltd.*, Case No. 4:20-cv-00279 (E.D. Tex.) (voluntarily dismissed).

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

Petition for *Inter Partes* Review
Patent No. 10,306,026

The '026 patent claims priority to U.S. Patent Application No. 15/694,802, filed on September 3, 2017, and issued as U.S. Patent No. 9,923,995 (“the '995 patent”). (EX1001, Cover.) The '995 patent is subject to the following instituted IPRs: *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845 (“Google -845 IPR”); and *Unified Patents, LLC. v. Jenam Tech, LLC*, IPR2020-00742 (“Unified -742 IPR”).

Petitioner is concurrently filing another IPR petition challenging the '026 patent.² Petitioner is also concurrently filing IPR petitions challenging U.S. Patent No. 10,069,945 which is also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.).

Petitioner has also filed the following IPR petitions challenging patents related to the '995 and '026 patents, and which are also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.):

- IPR2021-00627 (U.S. Patent No. 10,375,215, “the '215 patent”);
- IPR2021-00628 (U.S. Patent No. 10,075,564, “the '564 patent”);
- IPR2021-00629 (the '564 patent); and
- IPR2021-00630 (U.S. Patent No. 10,075,565, “the '565 patent”).

² Petitioner concurrently submits herewith its Notice Regarding Multiple Petitions.

Petition for *Inter Partes* Review
Patent No. 10,306,026

Counsel and Service Information: Lead counsel: Naveen Modi (Reg. No. 46,224). Backup counsel: (1) Joseph E. Palys (Reg. No. 46,508), (2) Quadeer Ahmed (Reg. No. 60,835), (3) Jason Heidemann (Reg. No. 77,880).

Service information is Paul Hastings LLP, 2050 M Street NW, Washington, DC 20036, Tel.: 202.551.1700, Fax: 202.551.1705, email: PH-Google-Jenam-IPR@paulhastings.com. Petitioner consents to electronic service.)

III. PAYMENT OF FEES

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-2613.

IV. GROUNDS FOR STANDING

Petitioner certifies that the '026 patent is available for review and Petitioner is not barred or estopped from requesting review on the ground identified herein.

V. PRECISE RELIEF REQUESTED

A. Claims for Which Review Is Requested

Petitioner requests review of claims 52, 55–57, 59–61, 64, 66–69 (“the challenged claims”) of the '026 patent and cancellation of those claims as unpatentable.

B. Statutory Ground of Challenge

The challenged claims should be canceled as unpatentable given the following ground:

Petition for *Inter Partes* Review
Patent No. 10,306,026

Ground 1: Claims 52, 55–57, 59, 61, 64, 66, and 68–69 are unpatentable under pre-AIA 35 U.S.C. § 103(a) over U.S. Publication No. 2007/0171921 to *Wookey et al.* (EX1005, “*Wookey*”) in view of U.S. Patent No. 6,674,713 to *Berg et al.* (EX1007, “*Berg*”); and

Ground 2: Claims 60 and 67 are unpatentable under pre-AIA 35 U.S.C. § 103(a) over *Wookey* in view of *Berg* and U.S. Patent No. 7,389,512 to Tucker (EX1031, “*Tucker*”).³

The ’026 patent issued from Application No. 16/040,517, filed July 19, 2018, which is a continuation of Application No. 15/915,047, filed March 7, 2018, which is a continuation of Application No. 15/694,802, filed September 3, 2017, which is a continuation-in-part of Application No. 14/667,642, filed March 24, 2015, which is a continuation-in-part of Application No. 13/477,402, filed May 22, 2012, which is a continuation of Application No. 12/714, 454 (“the ’454 application”), filed February 27, 2010. (EX1001, 1:8–34.)

³ Other references discussed herein are provided to show the state of the art at the time of the alleged invention of the ’026 patent.

Petition for *Inter Partes* Review
Patent No. 10,306,026

For purposes of this proceeding only, Petitioner assumes, without conceding, the earliest effective filing date of the '026 patent is February 27, 2010 (filing date of the '454 application).

Wookey published July 26, 2007, from an application filed November 14, 2006. (EX1005, Cover.) *Berg* issued January 6, 2004, from an application filed February 23, 1999. (EX1007, Cover.) *Tucker* published June 17, 2008, from an application filed January 27, 2004. (EX1031, Cover.) Thus, *Wookey*, *Berg*, and *Tucker* qualify as prior art at least under 35 U.S.C. § 102(b) (pre-AIA).

The above references were not considered during prosecution of the '026 patent. (EX1001 at Cover; *generally* EX1004.)

VI. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art at the time of the alleged invention of the '026 patent ("POSITA") would have an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking. (EX1002 ¶¶16–18.)⁴ More education can supplement practical experience and vice versa. (*Id.*)

⁴ Petitioner submits the declaration of Dr. Bill Lin (EX1002), an expert in the field of the '026 patent. (*Id.* ¶¶3–15; EX1003.)

VII. OVERVIEW OF THE '026 PATENT

The '026 patent is directed to networking and the sharing of information for detecting an idle TCP connection. (EX1001, 2:32–34, 8:37–46.) Figure 7 illustrates such a process.

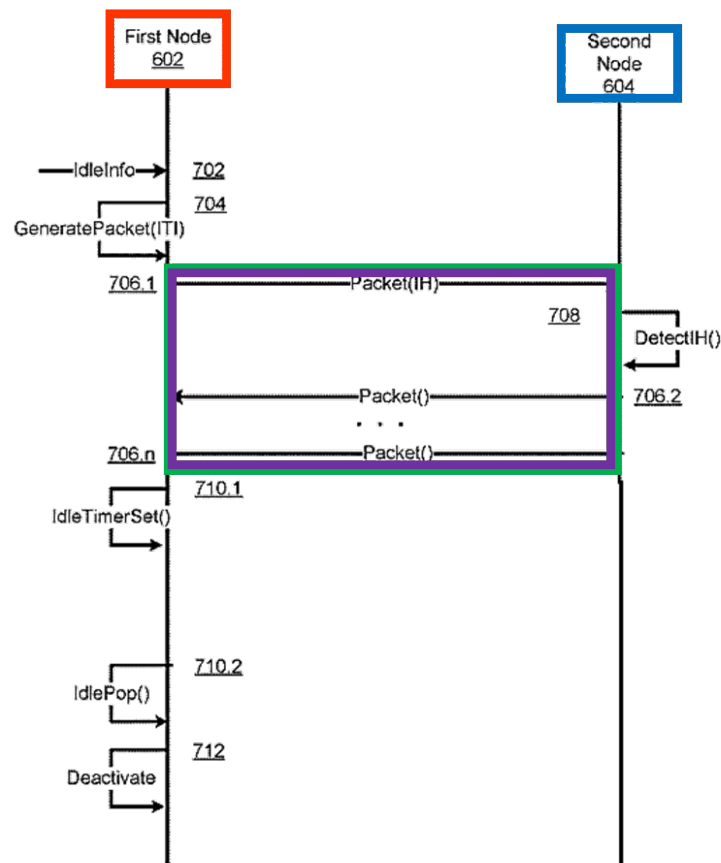


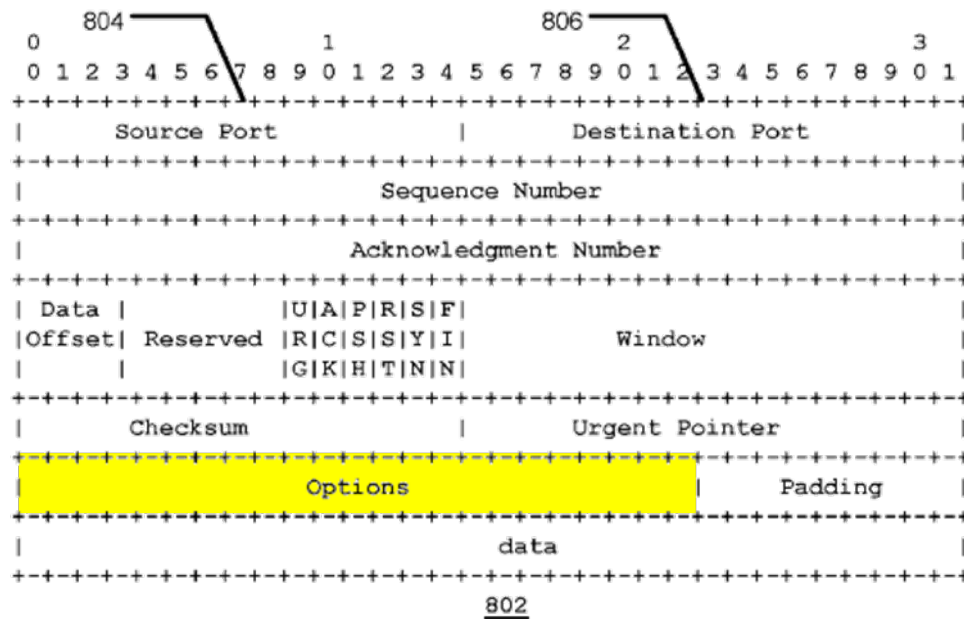
Fig. 7

(*Id.*, FIG. 7 (annotated); EX1002 ¶¶66.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

First, the **first node 602** receives a message 702 identifying idle information representing a duration for an idle time period (“ITP”). (EX1001, 11:62–64.) The idle information “may include and/or identify a duration of time for detecting an idle time period.” (*Id.*, 12:42–47.) The “duration may be specified according to various measures of time[,] including seconds.” (*Id.*; EX1002 ¶67.)

Next, the **first node** “generat[es] a TCP packet including an ITP header based on received idle information.” (*Id.*, 16:10–14.) The **TCP options field** of a TCP packet may store this ITP header. (*Id.*, 15:31–51.) Figure 8 below, “adapted from [IETF] RFC 793,” illustrates this configuration. (*Id.*, FIG. 8, 6:35–37, 15:31–51.)



Petition for *Inter Partes* Review
 Patent No. 10,306,026

The **TCP options field** contains a KIND sub-field that identifies the type of option presented, a length sub-field that specifies the length of the option field, and an **ITP Data sub-field** containing data. (EX1001, FIG. 8.)

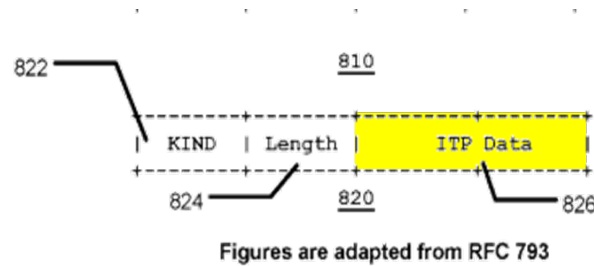


Fig. 8

(*Id.*, FIG. 8 (cropped and annotated); EX1002 ¶69.)

The ITP header is exchanged during the three-way handshake. (EX1001, 14:56–67.) For example, the **first node** transmits a message 706.1 (TCP packet) including an ITP header (IH) containing ITP information, to **second node 604**. (*Id.*, 16:35–36.) Message 708 exemplifies the **second node**’s detection of the IH in the received TCP packet. (*Id.*, 21:58–63; EX1002 ¶70.)

Based on the idle information, “a TCP keep-alive option, a TCP user timeout, a retransmission timeout, an acknowledgment timeout, and/or another timeout associated with a TCP connection may be modified.” (EX1001, 13:54–57; EX1002 ¶71.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

The challenged claims recite limitations relating to features discussed above. However, the limitations in the challenged claims were known in the prior art and obvious. (*See infra* Section IX; EX1002 ¶¶72, 100, 252; *see also id.* ¶¶19–65 (discussing technology background), citing, e.g., EXS. 1008–1009, 1011–1016, 1018–1023.)

VIII. CLAIM CONSTRUCTION

Under the applicable standard in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), claim terms are typically given their ordinary and customary meanings as understood by a POSITA at the time of the invention based on the claim language, specification, and the prosecution history of record. *Phillips*, 415 F.3d at 1313; *see also id.* at 1312–16. The Board, however, only construes the claims when necessary to resolve the controversy. *Toyota Motor Corp. v. Cellport Sys., Inc.*, IPR2015-00633, Paper No. 11 at 16 (Aug. 14, 2015) (citing *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)). Petitioner believes that no express constructions of any claim terms are necessary to assess whether the prior art reads on the challenged claims.⁵ (EX1002 ¶73.)

⁵ Petitioner reserves all rights to raise claim construction and other arguments, including challenges under 35 U.S.C. §§ 101 or 112, in district court as relevant to

Petition for *Inter Partes* Review
Patent No. 10,306,026

Claims 52 and 61 recite the terms “non-TCP packet” (EX1001, 27:44–48, 28:37–45) and claims 52, 59, 61, and 66 “non-TCP connection” (*e.g., id.*, 27:44–48, 28:20–23, 28:37–45, 29:55–59). Claims 68 recites the term “a second protocol, that is separate from the TCP.” (EX1001, 30:1–2.) The “non-TCP” terms are also recited in claims of the related ’995 patent at-issue in the Unified -742 IPR (*see supra* Section II). The ’026 patent and the ’995 patent share a common specification. (*Compare* EX1001 *with* EX1010 (’995 patent).)

The ’026 patent only discloses a second protocol, that is “separate from the TCP” term in the “Summary” section (EX1001, 3:7–52), which explains is “not an extensive overview of the disclosure and it does not identify key/critical elements ... or delineate the scope of the invention” (EX1001, 2:34–36). As discussed below, *Berg*’s RUDP protocol is a non-TCP protocol and is separate from the TCP, and the RUDP packets exchanged are non-TCP packets. (*See, e.g., infra* Sections IX.A.1.b–d, IX.A.9.c.) The Board preliminarily found that *Berg* discloses the non-TCP limitations. Unified -742 IPR, Paper 11 at 10–11.

those proceedings. *See, e.g., Target Corp. v. Proxicom Wireless, LLC*, IPR2020-00904, Paper 11 at 11–13 (November 10, 2020).

Petition for *Inter Partes* Review
Patent No. 10,306,026

Thus, because Petitioner relies on *Berg*'s similar disclosures to satisfy the non-TCP or separate from TCP terms herein (*see infra* Section IX.A), and given that the '026 patent offers no evidence requiring a special meaning of these terms and that the prior art discloses these features under any reasonable interpretation, construction of these terms is unnecessary.

IX. DETAILED EXPLANATION OF GROUND**A. Ground 1: The Combination of *Wookey* and *Berg* Render Obvious Claims 52, 55–57, 59, 61, 64, 66, 68–69****1. Claim 52****a) A method comprising:**

To the extent that the preamble of claim 52 is limiting, *Wookey* discloses the limitations therein. (EX1002 ¶¶101–109.)

Wookey discloses a method, as claimed. For example, *Wookey* teaches “a *method* for making a hypermedium page interactive.” (EX1005, Abstract; *see also id.* ¶[0002]; EX1002 ¶¶102, 74–83 (*Wookey* overview).)⁶

Figures 1, 2C, and 3D below show an exemplary environment in which a **client machine 10**, **remote machine 30**, and **remote machine 30'**, amongst other components, are employed to carry out the method. (*See also* EX1005 ¶¶[0016], [0019], [0020], [0024], [0026], [0192]–[0196], [0207], [0213].) These client and remote machines are “typical computers” (*id.* ¶[0021], FIGS. 1A–1B) including personal computers and wireless devices (*id.* ¶[0171]) such as mobile telephone or other portable telecommunication devices (*id.* ¶¶[0172]–[0173]).

⁶ Emphasis is added unless otherwise stated.

Petition for *Inter Partes* Review
 Patent No. 10,306,026

FIG. 1

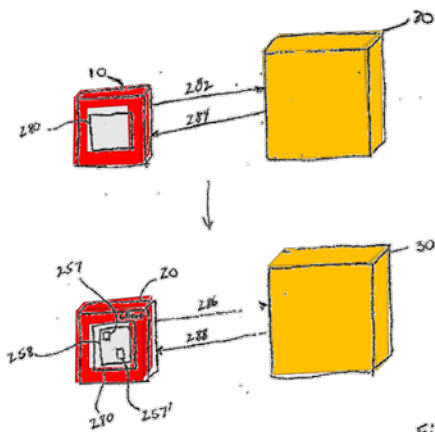
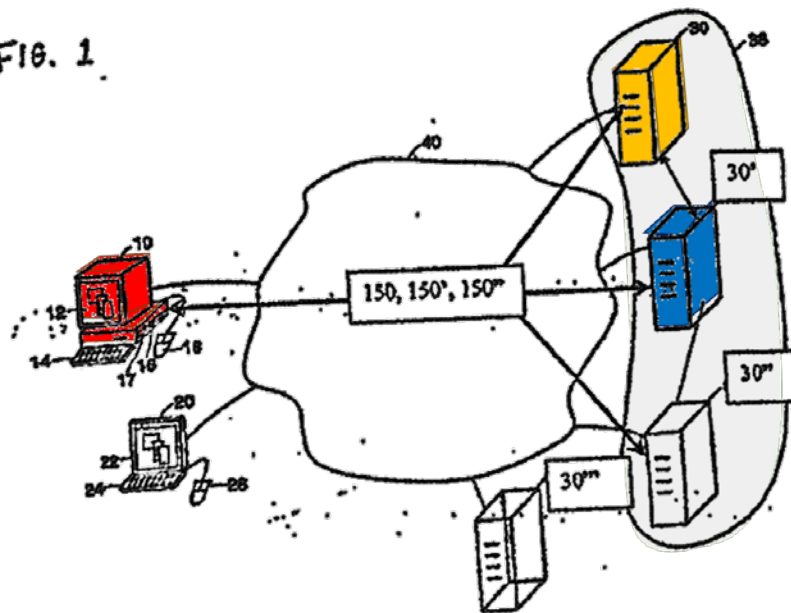


Fig. 2C

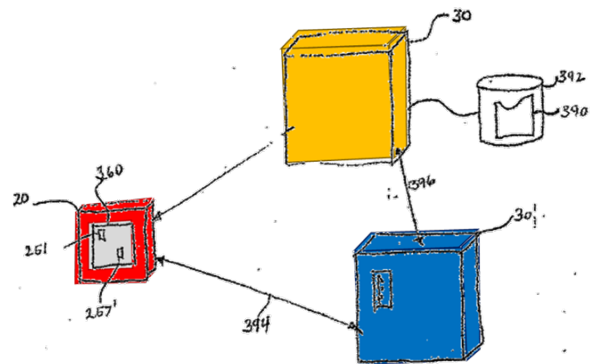


Fig. 3D

(EX1005, FIGS. 1, 2C, 3D (each annotated); EX1002 ¶¶103–106; *see also* EX1002 ¶107.)

Thus, *Wookey* discloses “[a] method,” as claimed. (*See also infra* Sections IX.A.1.b–f; EX1002 ¶¶108–109.)

b) **Claim 52.b**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation, which is addressed below in two parts. (EX1002 ¶¶110–149.)

- (1) **at a node configured to execute a network application such that the network application operates in accordance with a non-transmission control protocol (TCP) protocol []**

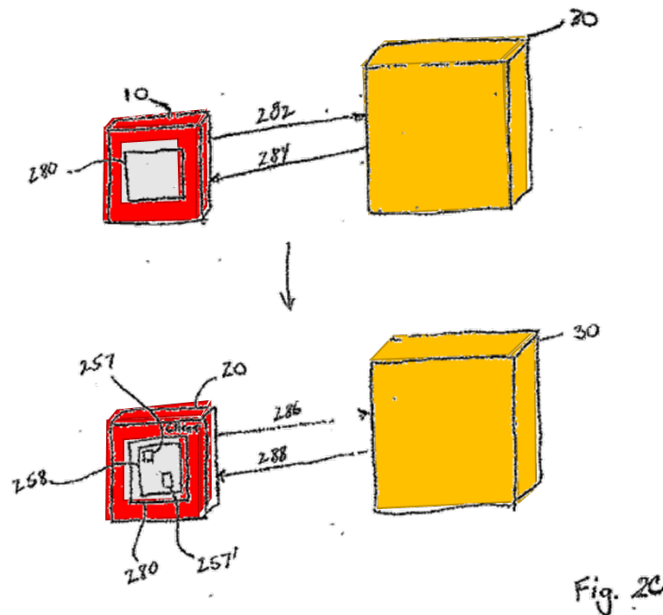
As explained below, the *Wookey-Berg* combination discloses and/or suggests **client machine 10** (“node”) is configured to execute a web browser application 280 (e.g., “network application”) such that the web browser application 280 operates in accordance with an RUDP protocol (“non-transmission control protocol (TCP) protocol”) to communicate with a different **remote machine 30’**. (See, e.g., EX1005 ¶¶[0192]–[0193], [0196], [0207], [0213], [0216]; *supra* Section IX.A.1.a; EX1002 ¶¶111.)⁷

For example, *Wookey* discloses, regarding Figure 2C, that **client machine 10** executes a web browser application 280 (e.g., a “network application”)

⁷ *Wookey* uses the term “machine” to describe what the ’026 patent refers to as a “node.” (EX1005 ¶¶[0153], [0301]; EX1002 ¶111 n.5.)

Petition for *Inter Partes* Review
 Patent No. 10,306,026

(EX1005 ¶[0192]) that “transmits a request 282 to access ... an HTML page residing on” **remote machine 30** (*id.* ¶[0193].) (*See also id.* ¶¶[0020], [0024], [0213].)⁸



(*Id.*, FIG. 2C (annotated); EX1002 ¶112.)

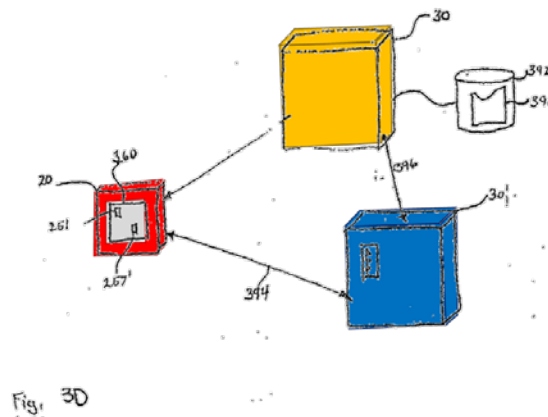
Remote machine 30, in turn, transmits an HTML page 288, which contains encoded URLs, to **client machine 10**. (EX1005 ¶[0193]; *id.* ¶¶[0207], [0213], [0754], [0757], FIG. 27; EX1002 ¶113.) Each encoded URL contains information

⁸ **Client machine 10** may use a TCP protocol for establishing a TCP connection with **remote machine 30**. (EX1005 ¶¶[0732], [0156]; *see infra* Section IX.A.1.f; EX1002 ¶112 n.6)

Petition for *Inter Partes* Review
Patent No. 10,306,026

for the **client machine 10** to create a connection to the remote machine hosting the resource. (EX1005 ¶[0216].) For example, the encoded URL specifies the location of the resource, a launch command associated with the resource. (*Id.*) Thus, when a user *clicks* an icon 257, 257' from the displayed HTML page 288, **client machine 10** utilizes the web browser 280 to use the encoded URL associated with the clicked icon to initiate the process for establishing a second connection (e.g., connection 394 in Figure 3D below) with **remote machine 30'**. (*Id.*) Accordingly, the **client machine 10** ("node") is configured to execute web browser application 280 ("network application"), based on the encoded URL's instructions when the URL is selected, such that the browser operates following a protocol to access resources from one or more remote machines. (EX1002 ¶113.)

Petition for *Inter Partes* Review
 Patent No. 10,306,026



(*Id.*, FIG. 3D (annotated); EX1002 ¶113.)

Thus, *Wookey* discloses, “at a node configured to execute a network application such that the network application operates” to establish a connection (e.g., the connection 394), as claimed. (EX1002 ¶114.)

Regarding the connection established between **client machine 10** and **remote machine 30'**, **client machine 10**'s web browser application 280, which is an “HTTP client agent,” “can use any type of protocol.” (EX1005 ¶[0159]; *see also id.* ¶[0155], [0192]–[0195].) For example, connection 394 can be made using various protocols, including the Virtual Network Computing (VNC) protocol. (*Id.* ¶[0216].) VNC is a desktop-sharing system that allows a computer to remotely access and control another computer. (EX1002 ¶115; EX1014 ¶[0019].) VNC can run over various transport protocols, including “TCP/IP, IPX/SPX, and NetBEUI.” (EX1005 ¶[0225]; *see also id.* ¶[0155], [0216].) IPX/SPX and NetBEUI protocols are

Petition for *Inter Partes* Review
Patent No. 10,306,026

examples of non-transmission control protocol (TCP) (“non-TCP”) protocols. (EX1002 ¶116; EX1013, 1:20–33.)

Wookey further explains that **client machine 10** may use “a different type of protocol than the one used to send the request to the remote machine 30.” (EX1005 ¶[0732]; EX1002 ¶118.)

Wookey also identifies desired characteristics associated at least with the second connection that a POSITA would have considered when choosing a protocol, such as negotiating parameters related to the connection, reducing unintentional termination of sessions due to an imperfect connection, detecting and handling disconnections (like when a mobile device enters an elevator), and specifying an inactive time before connection termination. (EX1002 ¶119; EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153].)

Wookey further explains that machines **10** and **30’** “exchange a set of messages which negotiate the parameters under which communications will occur.” (EX1005 ¶[0774].) Once negotiations are complete, **client machine 10** and **remote machine 30’** communicate over a network application. (*Id.*; see also *id.* ¶¶[0738]–[0739], [0744], [0772]–[0773]; EX1002 ¶120.)

Moreover, when **client machine 10** uses one of the encoded URLs (e.g., the one pointing to **remote machine 30’**) on the HTML page 288, it results in **client**

Petition for *Inter Partes* Review
Patent No. 10,306,026

machine 10 utilizing the web browser application to establish a connection (e.g., connection 394) with **remote machine 30'** using a transport protocol. (EX1002 ¶121.)

While *Wookey* discloses that various protocols, including non-TCP protocols (e.g., IPX/SPX), may be used to establish the second connection (EX1005, ¶[0225]) (including its desired characteristics), *Wookey* does not expressly provide details regarding establishing and maintaining the second connection or the types of negotiable parameters used to establish the second connection. (EX1002 ¶122.) However, it would have been obvious to consider and implement a protocol following *Wookey's* desired characteristics (e.g., such as those identified above) for the second connection between **client machine 10** and **remote machine 30'**, which may be “a different type of protocol than the one used [with] remote machine 30.” (EX1005 ¶[0732].)

Berg describes such a non-TCP protocol that allows nodes to negotiate or modify timeout timers on a per-connection basis to provide a reliable and efficient network connection. Accordingly, based on the teachings of *Berg*, a POSITA's knowledge, and guidance from *Wookey*, it would have been obvious to configure *Wookey's* **client machine 10** such that when the encoded URLs in the HTML page 288 is used, **client machine 10** uses the network application (browser application

Petition for *Inter Partes* Review
Patent No. 10,306,026

280) to operate in accordance with a protocol such as the one described by *Berg* having at least some characteristics contemplated by *Wookey* for such a second connection between **client machine 10** and **remote machine 30'**. (EX1002 ¶123.)

(a) *Berg*

Berg, like *Wookey*, relates to establishing a reliable connection between a client and server as it states that “it is important to be able to maintain multiple IP connections or sessions between the [devices], so that message[s] can be transmitted even when the inherently unreliable IP network fails” and that “[t]he system must be configured so that communications are not interrupted upon network failure, and so that communications can resume when the network comes back up.” (EX1007, 2:4–11; *see also id.*, 1:16–32, 1:54–64, 1:65–2:3, 2:43–47, FIGS. 1A–B; EX1002 ¶¶84–97 (*Berg* overview), 124.) Thus, *Berg* discloses networking features in the same technical field as *Wookey*. (EX1002 ¶125; *see also infra* Section IX.A.1.b.(1)(b) (discussing similarities between *Wookey* and *Berg*).)

Berg provides a gateway device (a client) connected to a media gateway controller (a server). (EX1007, 3:14–23, 2:55–60.) In *Berg*’s system, both the client and server execute a session manager to manage the network communications. (EX1007, 2:54–60, 7:63–8:2, 8:24–31.) The session manager operates “above a *reliable communication layer*” (*Id.*, 2:54–60), which “determines when or if a

Petition for *Inter Partes* Review
 Patent No. 10,306,026

session is connected or failed” (*id.*). “A protocol layer is ‘reliable’ when it guarantees that it will deliver or generate an error message for every message that an application program requests the transport layer using that protocol to transport.” (*Id.*, 8:27–31; *see also id.*, 8:11–13.) *Berg*’s Figure 2 below describes the layers, based on the Open Systems Interconnection (OSI) reference model, that computer systems use to communicate over a network. (*Id.*, 8:3–17, 5:27–55.)

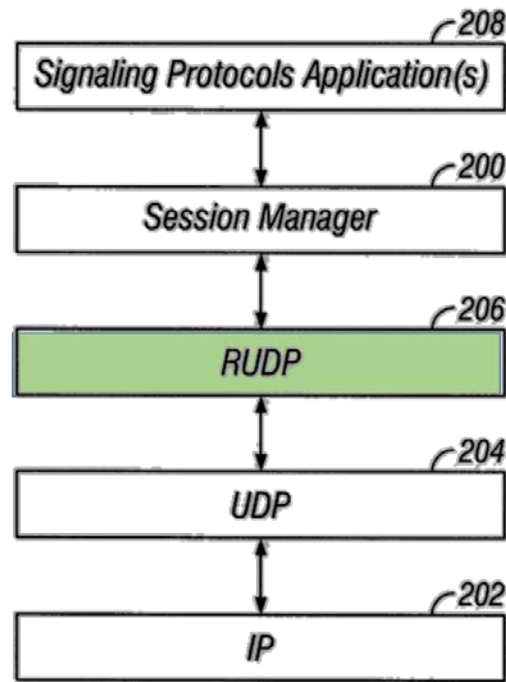


FIG. 2

(*Id.*, FIG. 2 (annotated); EX1002 ¶¶126–127.)

Berg explains that a **Reliable User Datagram Protocol (RUDP) 206** runs on top of the User Datagram Protocol (UDP) protocol software 204. (EX1007, 8:10–17.) The UDP is a non-TCP communications protocol that is primarily used for

Petition for *Inter Partes* Review
Patent No. 10,306,026

establishing low-latency and loss-tolerating connections, especially time-sensitive transmissions between applications. (EX1002 ¶¶128–129; EX1015 ¶[0110]; EX1016 ¶[0010].) The RUDP layer 206 allows *Berg*’s system to determine “when or if a session is connected or failed.” (EX1007, 2:54–60.)⁹ Above the **RUDP layer 206** in Figure 2 is the Session Manager 200 and Signaling Software Application(s) 208. (*Id.*, 8:24–27.) The Session Manager 200 can run above any reliable communication mechanism (e.g., RUDP 206). (*Id.*)

Berg notes that the “TCP/IP has a number of characteristics that make it unsuitable for” some applications. (*Id.*, 17:28–30.) In particular, “[m]ost TCP/IP implementations that allow properties like timers to be modified, *do not allow* the modification to be done on a per-connection basis.” (*Id.*, 17:40–42; EX1002 ¶130.)

In contrast to TCP/IP implementations, RUDP allows the “*characteristics of each connection to be individually configured* so that many protocols with different transport requirements can be implemented simultaneously on the same platform.” (EX1007, 17:42–46.) For example, RUDP allows the negotiation of various parameters including timeout values associated with the retransmission timeout (*id.*,

⁹ *Berg* uses “RUDP layer 206” and “communication layer” to refer to the same layer in Figure 2. (EX1002 ¶128 n.9; EX1007, 2:54–60, 8:13, 8:24–28.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

20:24–29, 22:44–56), acknowledgement timeout (*id.*, 20:29–35, 23:15–24), null segment (*id.*, 20:36–41, 23:45–59), and transfer state (*id.*, 20:42–47, 24:15–25). (EX1002 ¶131.)

Further, the “RUDP is a lightweight protocol layer designed to run on top of UDP [that] can provide reliable in-order delivery” (EX1007, 16:66–67; *see also id.*, 17:9–16), and it “has a *very flexible design* that would make it *suitable for a variety of transport uses*. (*Id.*, 17:13–15.) In an RUDP connection, *Berg* explains that the messages sent are UDP packets (not TCP packets) that include an RUDP header. (EX1007, 17:59–18:20, 19:1–40; EX1002 ¶¶131–132; *see also infra* Section IX.A.1.c.)

Thus, the RUDP is a non-TCP protocol. (EX1002 ¶133.) *See also* Unified - 742 IPR, Paper 11 at 10 (Board agreeing that *Berg*’s RUDP is a non-TCP protocol).

(b) *Reasons to Combine*

Based on *Wookey*'s and *Berg*'s disclosures, a POSITA's knowledge, and the discussions above, it would have been obvious to configure and implement *Wookey*'s **client machine 10** (including its web browser application 280) to use a protocol like *Berg*'s RUDP ("non-TCP protocol") to set up an RUDP ("second protocol") connection with **remote machine 30'** (*see, e.g.*, EX1005, Figure 3D). (EX1002 ¶134.) Such an implementation would have had many advantages and been a predictable and straightforward result achieved by combining well-known technologies using known methods. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 416–18 (2007).

As explained, both *Wookey* and *Berg* are directed to establishing a reliable connection between two nodes, where one or both nodes may experience an unintentional disconnection and thus disclose features in a similar technological field. (EX1005 ¶¶[0581], [1135]–[1136] (discussing the need for connections to manage unintentional disconnection), [1153]; EX1007, 1:16–32, 1:54–2:11, 2:43–47 (discussing importance of maintaining connections and handling disruptions); *supra* Section IX.A.1.b.1.) Thus, a POSITA would have had reason to consider *Berg* when contemplating and implementing the teachings of *Wookey*. (EX1002 ¶135.) And when collectively considered, such a POSITA would have been motivated to

Petition for *Inter Partes* Review
Patent No. 10,306,026

modify *Wookey*'s **client machine 10** to use a protocol like *Berg*'s RUDP to set up an RUDP connection with **remote machine 30'**. (*Id.*)

As discussed, *Wookey* describes a **client machine 10** ("node") that is configured to execute a web browser 280 ("network application") such that the browser establishes, when an encoded URL is selected, a second connection between **client machine 10** and **remote machine 30'** using a protocol that can be different from the one used to connect **client machine 10** and **remote machine 30**. *Wookey* further discloses the types of characteristics a POSITA would have considered concerning the protocol used for the second connection, such as negotiating parameters related to the connection, reducing unintentional termination due to an imperfect connection, detecting and handling disconnections, and specifying a permissible inactive time before connection termination. (*See supra* Section IX.A.1.b.1.) A POSITA would have been thus motivated by *Wookey*'s disclosures to consider protocols that would provide the desired features to *Wookey* for the second connection with **remote machine 30'**, including the features associated with the RUDP described by *Berg*. A POSITA would have been motivated to incorporate a protocol like *Berg*'s RUDP (a non-TCP protocol) in *Wookey*'s network application for establishing a connection between **client machine 10** and **remote machine 30'**. (EX1002 ¶136.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

A POSITA would have also recognized the benefits of using the RUDP protocol described in *Berg* with *Wookey*'s system/method, such as providing **client machine 10** with an improved reliable connection to **remote machine 30'** with the versatility of negotiated timeout parameters consistent with *Wookey*'s desired characteristics for the second connection. *See KSR*, 550 U.S. 416–17. Indeed, by using a protocol like *Berg*'s RUDP protocol, *Wookey*'s method would enable negotiation of properties like the timeout values on a per-connection basis over a reliable connection (EX1007, 24:40–47) and provide alerts to the nodes if the session fails (*id.*, 2:55–60). (EX1002 ¶137.) *Wookey*'s method for “providing a client with a reliable connection to a host service” (EX1005 ¶[0077]) would therefore have been improved by providing a connection that is “very flexible ... [and] suitable for a variety of transport uses,” as provided by *Berg* (EX1007, 17:9–16, 17:42–46.). (EX1002 ¶137.)

It would have been a foreseeable and straightforward implementation to configure *Wookey*'s **client machine 10** such that at least one of the encoded URLs in HTML page 288, when used, would cause **client machine 10** (e.g., by way of its web browser 280) to operate following *Berg*'s RUDP (“non-TCP protocol”). (EX1002 ¶138.) *See KSR*, 550 U.S. at 416. For example, as noted above, *Wookey* discloses that “*the URL includes a file, or a reference to a file, that contains the*

Petition for *Inter Partes* Review
Patent No. 10,306,026

information necessary for the client to create a connection to the remote machine hosting the resource.” (EX1005 ¶¶0216].) In light of the reasons above, a POSITA would have been motivated to configure *Wookey*’s encoded URL such that it includes information resulting in **client machine 10** utilizing the web browser application 280 to operate in accordance with a non-TCP protocol (like *Berg*’s RUDP) to initiate a process of establishing a second connection with **remote machine 30**’. (EX1002 ¶¶138.) This implementation would have been achieved by combining well-known technologies using known methods, such as known network design concepts described by *Wookey* and *Berg*, and known in the art. (*Id.*)

Berg’s protocol is flexible and suited for various uses and allows the nodes to negotiate properties like the timeout values on a per-connection basis over a reliable connection. (EX1007, 24:40–47.) These are features that *Wookey* itself recognizes (EX1005 ¶¶0721]) and are features that would have improved *Wookey*’s process, like as described in *Berg*. Accordingly, such a combination would have yielded the predictable result of providing a network communication exchange that minimizes disruptions caused by timeout issues. (EX1002 ¶139); *see KSR*, 550 U.S. at 417.

The *Wookey-Berg* combination would have involved using known technologies (e.g., aspects of similar protocols) and design concepts and processes to obtain the foreseeable result of a reliable connection between *Wookey*’s **client**

Petition for *Inter Partes* Review
Patent No. 10,306,026

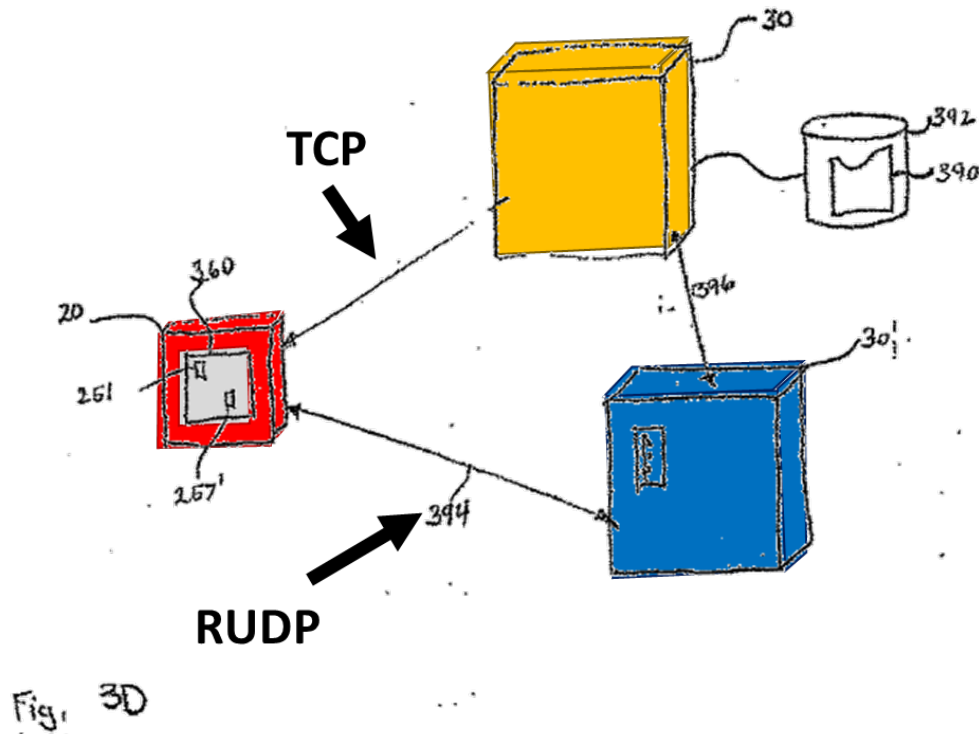
machine 10 and **remote machine 30'**. (EX1002 ¶140.) Indeed, the above-modification would have involved substituting features from one reliable protocol from a finite number of available alternative reliable communication protocols, such as that described by *Berg*. (*Id.*) Thus, a POSITA would have had a reasonable expectation of success in the above-modification. (*Id.*) See *Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1364 (Fed. Cir. 2007) (“only a reasonable expectation of success, not a guarantee, is needed” in an obviousness analysis).¹⁰

A POSITA would have been skilled and knowledgeable about configuring the above modification in various ways, while taking into account any known programming, design, and other related concepts, limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. (EX1002 ¶141.) For example, a POSITA would have been motivated based on such

¹⁰ There is no evidence of nonobviousness at this time that would overcome the overwhelming evidence of obviousness as presented herein. Petitioner reserves the right to address any such evidence/argument should PO later present any in this proceeding.

Petition for *Inter Partes* Review
 Patent No. 10,306,026

disclosures to configure *Wookey's* system and process consistent with the non-limiting example reflected below.¹¹



(EX1005, FIG. 3D (annotated); EX1002 ¶141.)

* * *

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 52.b(1). (EX1002 ¶142; *see also infra* Section IX.A.1.b.2, IX.A.1.c–f.)

¹¹ The configuration discussed below is exemplary and not limiting. (EX1002 ¶141.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

- (2) **[the non-TCP protocol] that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer:**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious limitation 52.b(2). (EX1002 ¶¶143–149.) As discussed below, **client machine 10** (“node”) in the *Wookey-Berg* combination includes a web browser application 280 (“network application”) that is configured to operate in accordance with the RUDP protocol (“non-TCP protocol”) that would have operated above an Internet Protocol (IP) layer and below a Hypertext Transfer Protocol (HTTP) application layer, as claimed.

For example, **client machine 10** “can access a resource ... displayed in the Resource Neighborhood web page” (EX1005 ¶[0216]), which is executed via the web browser application 280 (*id.* ¶[0192]) over the World Wide Web. *Wookey’s* web browser application 280 initiates requests and accesses resources according to HTTP (EX1005 ¶[0155]), a known communication protocol at the application layer

Petition for *Inter Partes* Review
Patent No. 10,306,026

of an OSI model or TCP/IP model¹² for the World Wide Web. (EX1002 ¶144; EX1018, 7–8, 12.)

Indeed, the communications link between the remote and client machines may use an “application layer protocol, such as the Hypertext Transfer Protocol (HTTP).” (EX1005 ¶[0155].) Further, **client machine 10** includes an “HTTP client agent,” such as a web browser, which “can use any type of protocol.” (*Id.* ¶[0159]; EX1002 ¶145.)

Thus, the RUDP in the *Wookey-Berg* combination would have operated below the application layer on which the HTTP operates. (EX1002 ¶146.) This is because, in the *Wookey-Berg* combination, the RUDP would have run on top of the User Datagram Protocol (UDP) protocol software 204 at a transport layer, and information exchanged via the RUDP between the client and server would have been communicated to the application layer where the HTTP-based web browser operated. (*Id.*) The transport layer is below the application layer. (*Id.*) Similarly,

¹² For purposes of this IPR, regardless of the model (OSI or TCP/IP) considered at the time, a POSITA would have had the same understanding as explained herein. (EX1002 ¶144 n.11.)

the RUDP in the *Wookey-Berg* combination would have operated above an IP layer because the transport layer operates above the IP layer. (*Id.*)

Berg confirms this understanding. (EX1007, 8:3–8; *see supra* Section IX.A.1.b.1.a.) As shown below, the **RUDP layer 206** operates above **IP layer 202** and below **Signaling Protocols Application(s) layer 208** (an application layer, where the HTTP would operate, i.e., “a hypertext transfer protocol (HTTP) application layer”). (EX1007, FIG. 2, 8:3–18, 8:24–31; *see also id.*, 1:15–59, 5:27–55.)

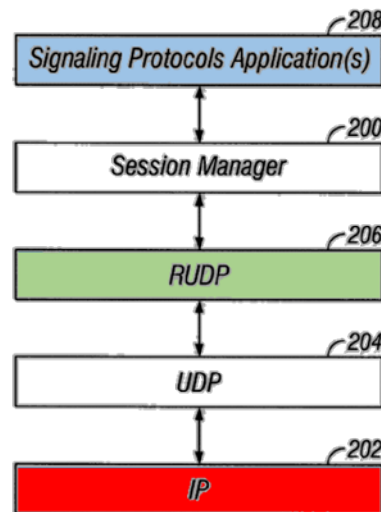


FIG. 2

(EX1007, FIG. 2 (annotated); EX1002 ¶147.)

Thus, the combination of *Wookey-Berg* discloses and/or suggests and renders obvious a **client machine 10** that includes a web browser application 280 (“network application”) that is configured to operate in accordance with the RUDP (non-TCP

Petition for *Inter Partes* Review
Patent No. 10,306,026

protocol) “that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer,” as claimed in limitation 52.b(2). (EX1002 ¶148.)

* * *

Accordingly, the *Wookey-Berg* combination discloses and/or suggests limitation 52.b. (EX1002 ¶149.)

- c) **receiving, from another node, a non-TCP packet during a setup of a non-TCP connection;**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶150–153.) As explained below, **client machine 10** in the combined *Wookey-Berg* system/method would utilize the network browser 280 to operate in accordance with the RUDP protocol to receive, from **remote machine 30'** (“another node”), a responsive SYN segment (“non-TCP packet”) during a setup of the RUDP connection (“non-TCP connection”). (*Id.*)

For example, when a client initiates an RUDP connection, it sends a SYN message to the server that contains the negotiable parameters. (EX1007, 24:40–47; *see also id.*, 18:56–62.) During the negotiation of the RUDP connection, the client will receive a responsive SYN segment (to its sent SYN message) from the server. (*Id.*, 24:40–47.) The server’s responsive SYN segment will either echo the client’s proposed parameters or propose different parameters. (*Id.*; EX1002 ¶151.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

The RUDP SYN messages/segments sent to set up the RUDP connection are UDP packets that include an RUDP header. (EX1007, 17:59–18:20; *see also id.*, 19:1–40.) RUDP messages are non-TCP packets because they are communicated using RUDP, which is a non-TCP protocol. (*Supra* Section IX.A.1.b.1.a.) Therefore, the responsive SYN segment sent from **remote machine 30'** and received by **client machine 10** in the *Wookey-Berg* combination is a non-TCP packet. (EX1002 ¶152; *supra* Section VIII.) Furthermore, the connection established according to an RUDP protocol is a non-TCP connection. (EX1002 ¶152; *supra* Section IX.A.1.b.1.a.)

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 52.c. (EX1002 ¶153.)

- d) **identifying metadata, that specifies at least one of a number of seconds or minutes, in an idle time period parameter field in the non-TCP packet, for an idle time period, where, as a result of a detection of the idle time period, the non-TCP connection is subject to deactivation; and**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶154–172.) The combined *Wookey-Berg* system/method discloses and/or suggests that **client machine 10** identifies a null segment timeout value (“metadata”), that specifies a number of seconds, in the null segment timeout parameter field in the SYN segment (“non-TCP packet”), for an idle time period,

Petition for *Inter Partes* Review
Patent No. 10,306,026

where, as a result of a detection of the idle time period, the RUDP (“non-TCP”) connection is subject to deactivation. (*Id.* ¶155; *see supra* Section IX.A.1.c.)

Berg discloses identifying the negotiable parameters in the SYN segment (“non-TCP packet”) for establishing an RUDP connection. For example, when the client receives the SYN segment sent from the server, it can *accept* or *reject* the proposed parameters (such as the null segment timeout value) in the received SYN segment. (EX1007, 24:39–48.) Thus, the client necessarily identifies each parameter in the SYN segment to decide whether it will accept or reject the proposed parameters. (EX1002 ¶156.) The *Wookey-Berg* combination would have thus been configured such that **client machine 10** identifies data (“metadata”) in the parameter fields in the RUDP SYN message (“non-TCP packet”). (*Id.* ¶157.)

To the extent the *Wookey-Berg* does not disclose the claimed “identify[ing],” it would have been obvious to modify the functionality in the combined *Wookey-Berg* system/process such that it identifies the negotiable parameters as claimed. (*Id.* ¶158.) A POSITA would have been motivated to configure the *Wookey-Berg* system/process to include functionality that identifies the parameters in the SYN segment to facilitate the subsequent process of accepting or rejecting the parameters in a manner consistent with the features described by *Berg*. (EX1007, 24:39–48; EX1002 ¶¶158–159.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

For example, Figure 2, below, of the RUDP specification,¹³ illustrates the SYN segment with all of its negotiable parameters for the RUDP connection:



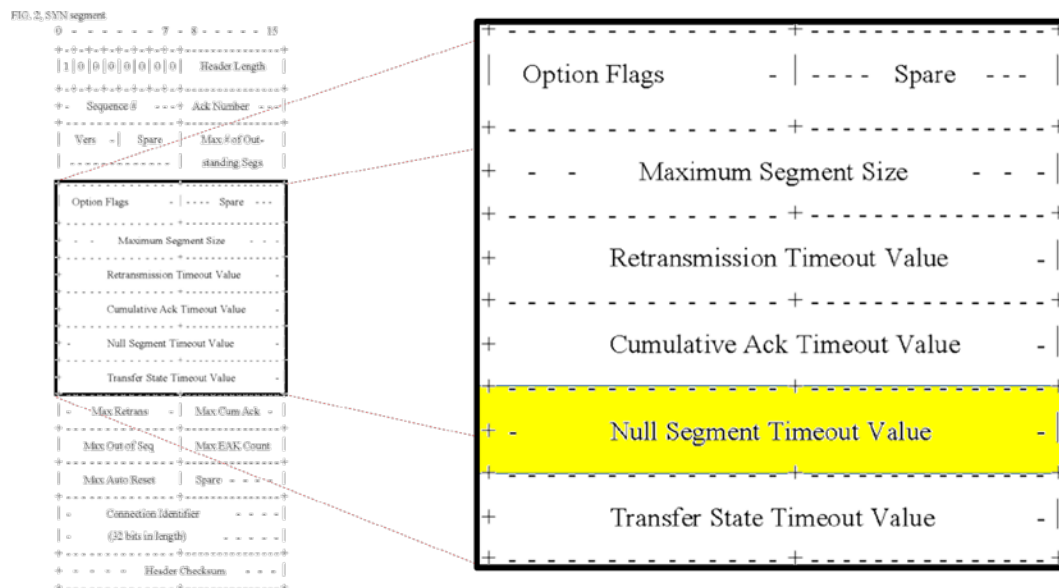
(EX1007, 19:1–40; EX1002 ¶160.)

¹³ The RUDP specification is included in *Berg* as “Appendix 1.” (EX1007, 16:59–25:54; *see also* EX1011, 6; EX1002 ¶160 n.12.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

The *Wookey-Berg* combination would have thus been configured such that **client machine 10** identifies data (“metadata”) in the parameter fields in the RUDP SYN segment (“non-TCP packet”). (EX1002 ¶161.)

Berg further provides that the null segment timeout value is specified in milliseconds (“a number of seconds”) and is identified from a null segment timeout value (“idle time period”) parameter field in an RUDP SYN segment (“non-TCP packet”). (E.g., EX1007, 18:58–60, 19:22–25, 20:36–42, 22:25–33; see also EX1002 ¶162.) Figure 2 of the RUDP specification illustrates the null segment timeout value parameter field that specifies the **null segment timeout value**:



(*Id.*, 19:1–40 (excerpted and annotated); see also EX1002 ¶163.)

This null segment timeout value (“metadata”) indicates the amount of time that a client will wait (“idle time period”) before sending a null segment when the

Petition for *Inter Partes* Review
Patent No. 10,306,026

connection has been idle (e.g., when a data segment has not been sent or received). (EX1007, 20:36–41, 22:25–33, 20:36–41.) Thus, this period of waiting by the client (when the connection is idle) before the client sends a null segment to the server to verify whether the connection is active represents an “idle time period,” as claimed. (EX1002 ¶164.)

This understanding that timeout values are metadata is confirmed by the ’026 patent, which explains that the metadata can be a value indicating a “duration of time” (EX1001, 20:58–64), which is what the timeout values in *Berg* contain (EX1007, 20:36–47; EX1002 ¶165.)

Furthermore, as a result of detecting a null segment timer’s expiration, which is set based on the null segment timeout value, the RUDP connection is “subject to deactivation,” as claimed. (EX1002 ¶166.) In particular, at the client, the null segment timeout value is used to set the duration of a null segment timer “for sending a null segment *if a data segment has not been sent.*” (EX1007, 20:36–42, 23:45–49, 22:25–32.)¹⁴ When the client detects the null segment timer has expired, it sends a null segment to the server to determine if the connection is still valid. (*Id.*, 23:45–

¹⁴ *Berg* uses “null segment,” “NUL segment” and “keep-alive” interchangeably. (EX1007, 20:36–42, 22:25–33; EX1002 ¶166 n.13.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

59.) When the server receives a null segment, it “*must* acknowledge the segment if a valid connection exists.” (*Id.*, 22:27–30.) Conversely, if the null segment is not received or acknowledged, the connection is subject to being closed via an “auto reset.” (*Id.*, 23:45–59; EX1002 ¶166.)

The “auto reset,” which is triggered upon the null segment timer’s expiration, is a form of deactivation of the RUDP connection. (EX1002 ¶167; EX1007, 21:15–23, 23:60–24:4.) For example, an “auto reset” signals a problem with the connection (e.g., connection failure) and that the connection needs to be deactivated. (EX1007, 23:60–24:4; *see also id.*, 21:15–23, 24:15–25.) When an auto-reset occurs at the client side, the client will send a reset (RST) segment to the server to close or reset the connection. (*Id.*, 23:60–24:4; *see also id.*, 18:27–28, 22:20–24; EX1002 ¶167.) Either closing or resetting a connection is a form of deactivation for the connection. (EX1002 ¶¶167–170.)

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been further configured to include the process of **client machine 10** identifying the null segment timeout value (“metadata”), that specifies a number of seconds or minutes, in the null segment timeout value parameter field (“idle time period parameter field”) in the RUDP SYN segment (“non-TCP packet”), for an idle time period, where, as a result of a detection of the

Petition for *Inter Partes* Review
Patent No. 10,306,026

idle time period, the RUDP (“non-TCP”) connection is subject to deactivation, as claimed. (EX1002 ¶¶171–172; *see supra* Section IX.A.1.b.)

e) **determining, based on the metadata, a timeout attribute associated with the non-TCP connection;**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶173–178.) **Client machine 10** in the *Wookey-Berg* combination determines a timeout attribute, e.g., the configuration of a null segment timer, based on the null segment timeout value (“metadata”) associated with the RUDP connection (“non-TCP connection”). (*Id.* ¶173; *supra* Section IX.A.1.d.)

For example, in an RUDP implementation, the client determines, based on the null segment timeout value (“metadata”), whether the null segment mechanism for the established RUDP connection is enabled (a “timeout attribute associated with the non-TCP connection”). (EX1007, 23:58–59; EX1002 ¶¶174–175.)

This understanding that the timeout attribute can be an attribute of a keep-alive mechanism is confirmed by the ’026 patent, which explains that, based on the idle time period specified by the metadata, “an attribute of a TCP keep-alive option” may be modified by “activat[ing], disabl[ing], and/or modify[ing] the state of the keep-alive option” (EX1001, 13:58–14:10; *see also id.*, 16:43–61 (describing that the “keep-alive timeout attribute” is set “based on a duration of” the idle time information specified in the metadata). 7:44–48, 28:37–45). (EX1002 ¶176.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

Accordingly, for reasons similar to those explained above, the combined *Wookey-Berg* process would have been further configured to include **client machine 10** determining, based on the null segment timeout value (“metadata”), a null segment timer’s enablement (“timeout attribute”) associated with the RUDP (“non-TCP”) connection. (EX1002 ¶¶177–178; *see also* Sections IX.A.1.b, IX.A.1.d.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

- f) **wherein, based on operating in accordance with the TCP protocol, a three-way TCP handshake is performed for establishing a TCP connection that is different than the non-TCP connection.**¹⁵

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶179–185.) For example, **remote machine 30** in *Wookey* causes the HTML page 288, which includes the encoded URLs, to be sent to a **client machine 10** over a TCP/IP connection. (*Supra* Section IX.A.1.b.) Thus, **client machine 10**, via its network application, operates according to the TCP protocol and performs a three-way TCP handshake for establishing a TCP connection with **remote machine 30** that is different from the RUDP (“non-TCP”) connection with **remote machine 30’**. (EX1002 ¶179.)

A three-way handshake is necessarily performed to establish a TCP connection. (EX1002 ¶180.) RFC 793, the TCP specification¹⁶, provides that a

¹⁵ The ’026 patent specification does not provide any disclosure describing the claimed “establishing a TCP connection that is different than the non-TCP connection” features. (EX1002 ¶179 n.15.) Therefore, while Petitioner addresses this limitation with respect to the asserted prior art, Petitioner does not concede that the ’026 patent provides adequate disclosures of such claimed features.

Petition for *Inter Partes* Review
Patent No. 10,306,026

three-way handshake is necessary (EX1008, 32) and that it is used to establish a TCP connection (*id.*, 34.) (*See also id.*, 31–32, 34–37; EX1002 ¶180.) The '026 patent acknowledges that the three-way handshake is required to establish a TCP connection, citing RFC 793. (EX1001, 14:56–15:3; EX1002 ¶181.)

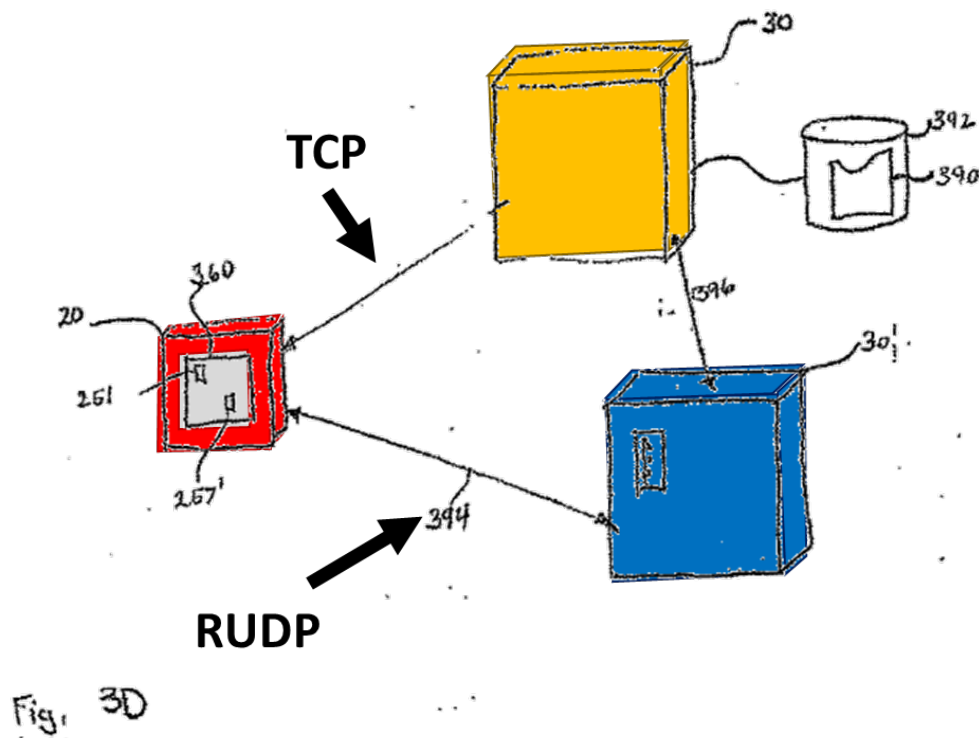
Thus, to establish a TCP connection between **client machine 10** and **remote machine 30**, **client machine 10** utilizes the web browser 280, when following the TCP protocol, to perform a three-way TCP handshake with **remote machine 30** for setting up a TCP connection between the machines, as claimed. (EX1002 ¶182.)

This TCP connection between **remote machine 30** and **client machine 10** is different than the RUDP (“non-TCP”) connection because it is separate from the RUDP connection between **remote machine 30'** and **client machine 10** in the *Wookey-Berg* combination. (*Id.* ¶183; *supra* Section IX.A.1.b.) Indeed, as discussed for limitation 52.b, *Wookey* explains that **client machine 10** may use a different protocol than the one used to send the request to **remote machine 30** when establishing the second connection with **remote machine 30'**. (*Supra* Section IX.A.1.b.) The **client machine 10** in the *Wookey-Berg* combination, as shown in

¹⁶ RFC 793 is incorporated by reference in the '026 patent (EX1001, 1:56–62, 11:39–40). (EX1002 ¶180 n.16.)

Petition for *Inter Partes* Review
 Patent No. 10,306,026

Figure 3D, establishes an RUDP-based connection with **remote machine 30'** that utilizes a *different* protocol than the TCP protocol used for establishing the first connection between **client machine 10** and **remote machine 30**. (EX1002 ¶184.) An exemplary arrangement relating to the above-discussed combination is exemplified below for Figure 3D of *Wookey*.¹⁷



(EX1005, FIG. 3D (annotated); EX1002 ¶¶184–185.)

¹⁷ See n.11.

2. Claim 55

- a) **The method of claim 52 wherein one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶186–191.)

As discussed above, **client machine 10** determines whether the null segment mechanism is enabled. (*Supra* Section IX.A.1.e.) If enabled, **client machine 10** further determines the timeout duration for a null segment timer associated with the established RUDP connection. (EX1007, 20:36–41, 23:45–59¹⁸; EX1002 ¶186; *supra* Section IX.A.1.d.).

Berg further discloses that the client will send a null segment (“one or more keep-alive packets”) to the server when its null segment timer expires. (EX1007, 20:36–41; 23:45–59.) The duration (“keep-alive period”) for the null segment timer is set proportionally to the identified null segment timeout value (metadata specifying the “idle time period”) obtained from the received RUDP SYN segment when the null segment mechanism is enabled. (EX1007, 20:36–38, 23:46–59; *supra* Sections IX.A.1.d–e.) For example, the duration of null segment timer for the client

¹⁸ See n.14.

Petition for *Inter Partes* Review
Patent No. 10,306,026

is set to one times that of the identified null segment timeout value—i.e., the null segment timer’s duration mirrors the “idle time period” specified by the identified null segment timeout value. (EX1007, 20:36–38, 23:46–59; EX1002 ¶188; *see supra* Section IX.A.1.d–e); *see also* Unified -742 IPR, Paper 11 at 9 (finding that *Berg*’s null segment timeout value teaches a time period for sending a null segment at the client side), 12, 17 (finding that *Berg*’s null segment timeout value teaches an idle time period).)

The above understanding is consistent with the ’026 patent’s disclosure, which provides that the keep-alive period can be “*configure[d] ... to expire based on the identified duration*” (EX1001, 19:14–15) and/or “activate[d], disable[d], and/or modif[ied]” based on the idle time period (*id.*, 14:3–11, 22:39–42), which is what *Berg* does. (*See also id.*, 16:43–46.)

Accordingly, in the *Wookey-Berg* combination, **client machine 10** communicates a null segment (“one or more keep-alive packets”) to **remote machine 30**, based on a null segment timer’s duration, which is in turn, based on the identified null segment timeout value (metadata specifying the “idle time period”) in the received SYN packet. (EX1002 ¶189–191; *supra* Sections IX.A.1.d–e.)

3. Claim 56

- a) **The method of claim 55 wherein the keep-alive period is not configurable, while the idle time period is configurable.**¹⁹

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶192–195.) As discussed above, the duration of the client’s null segment timer in the *Wookey-Berg* combination is set proportionally to the negotiated null segment timeout value obtained from the received SYN segment, when the null segment mechanism is enabled. (*Supra* Section IX.A.2; *see also supra* Sections IX.A.1.d–e.) As explained below, the null segment timeout value (metadata specifying the “idle time period”) is configurable, while the null segment timer’s duration (“keep-alive period”) is not.

For example, the “idle time period” in the *Wookey-Berg* combination is configurable because the client can negotiate the null segment timeout value, which

¹⁹ The ’026 patent specification does not provide any disclosure describing how the “keep-alive period is not configurable” given that the “keep-alive period” is “based on the idle time period” (claim 55) which “is configurable” (claim 56). (EX1002 ¶192 n.20.) Therefore, while Petitioner addresses such features with respect to the

Petition for *Inter Partes* Review
Patent No. 10,306,026

specifies the “idle time period,” with the server. (*See, e.g., supra* Sections IX.A.1.c–e.) In contrast, the client’s null segment timer’s duration is set to the period specified by the null segment timeout value once the connection is established, i.e., the client’s null segment timer’s duration is set to a fixed proportion of the period specified by the null segment timeout value (one times the null segment timeout value). Thus, it is not configurable by the client as the fixed proportion itself is not configurable during the connection setup.²⁰ (*See supra* Sections IX.A.2; EX1007, 20:36–38, 23:46–59; EX1002 ¶¶193.) This is consistent with PO’s view of these features. (EX1017, 43 (PO alleging that “[b]ased on the list of negotiable parameters ..., the keep-alive period is not configurable during connection setup,” and explaining that this keep-alive period is “constrained by the idle time period, which is negotiated at connection setup.”).)

asserted prior art, Petitioner does not concede that the ’026 patent provides adequate disclosures of such claimed features, or any other claimed features.

²⁰ Similarly, the server’s null segment timer’s duration is set to a fixed proportion relative to the null segment timeout value (two times the null segment timeout value). (EX1007, 23:51–52; EX1002 ¶193 n.21.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

Accordingly, in the *Wookey-Berg* combination, the null segment timer's duration is fixed ("not configurable"), while the "idle time period" specified by the negotiated null segment timeout value is configurable. (EX1002 ¶194–195.)

4. Claim 57

- a) **The method of claim 55 wherein the one or more keep-alive packets are conditionally communicated based on at least one keep-alive attribute.**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶196–197.) For example, the *Wookey-Berg* combination discloses and/or suggests this limitation for at least the same reasons discussed above for claim 55, where a null segment ("keep-alive packet") is conditionally communicated based on the expiration of the **client machine 10**'s null segment timer ("keep-alive attribute"). (*Supra* Section IX.A.2.). Indeed, as *Berg* discloses, "*[i]f the client's null segment timer expires*, [then] the client will send a null segment to the server." (EX1007, 23:45–49; EX1002 ¶¶196–197; EX1011, 13–14.)

5. Claim 59

- a) **The method of claim 55 wherein the node determines, based on the metadata, the timeout attribute associated with the non-TCP connection, in connection with negotiating, with the another node, a duration of the idle time period.**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶198–199.) For example, as discussed above, the **client machine 10** in the *Wookey-Berg* combination determines, based on the identified null segment timeout value (“metadata”), the **client machine 10**’s null segment timer’s enablement (“timeout attribute”) associated with the RUDP (“non-TCP connection”) (*supra* Sections IX.A.1.d–e, IX.A.2), in connection with negotiating, with the **remote machine 30**’ (“another node”), a duration of the null segment timeout value (“idle time period”) (*supra* Sections IX.A.1.b–e.) (EX1002 ¶¶199.)

6. Claim 61

- a) **A method comprising:**

To the extent limiting, *Wookey* discloses this limitation for at least the same reasons as above for limitations 52.a–f, explaining *Wookey*’s disclosure of a method for performing steps related to those recited in claim 61. (*See supra* Sections IX.A.1.a–f; EX1002 ¶200; *infra* Sections IX.A.6.b–f.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

- b) **at a node configured to execute a network application such that the network application is [sic] operates in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer:**²¹

The *Wookey-Berg* combination discloses and/or suggests this limitation for at least the same reasons above for limitation 52.b. (*Supra* Section IX.A.1.b; EX1002 ¶201.)

- c) **receiving idle information for use in detecting an idle time period that results in a non-TCP connection being subject to deactivation;**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶202–212.) As discussed for limitation 52.b, the above-discussed *Wookey-Berg* combination would have included processes that involved establishing an RUDP connection between **client machine 10** and **remote machine 30’**. (*Supra* Section IX.A.1.b.) As discussed above for limitation 52.c and further below, *Berg*’s RUDP protocol includes processes for receiving a null segment timeout value (“idle information”) for use in detecting an idle time period that results

²¹ Although the limitation recites “the network application *is* operates,” a POSITA would have understood that this is a typographical error and should have stated “the network application operates.” (EX1002 ¶201 n.22.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

in the RUDP (“non-TCP”) connection being subject to deactivation. (*Supra* Section IX.A.1.c–d, EX1002 ¶202.)

For example, when setting up a RUDP, the nodes will negotiate a null segment timeout value. (*Supra* Section IX.A.1.d.) As explained above, the null segment timeout value, which is “information,” corresponds to detecting an idle time period that results in the RUDP connection being subject to deactivation. (*Id.*) Accordingly, the null segment timeout value is “idle information.” (EX1002 ¶203.)

This understanding of “idle information” is consistent with the ’026 patent, which provides that “[i]dle information ... may include ... a duration of time ... [which] may be specified according to various measures of time including seconds.” (EX1001, 12:43–47; EX1002 ¶204.)

When the client initiates a connection, the client’s Upper Layer Protocol (ULP), e.g., application, receives negotiable parameters that indicate the client’s desired features for the RUDP connection being established. (EX1007, 24:40–48; *see also id.*, 18:57–64.) Because the client must select and include the desired negotiable parameters (e.g., null segment timeout value) in the SYN segment to be sent to the server, the client receives these negotiable parameters, e.g., null segment timeout value (“idle information”), *prior* to generating and sending the SYN segment. (EX1002 ¶205.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

Also, the client receives these parameters when the server initiates the connection. For example, when the server initiates a reset (EX1007, 23:61–65), the server sends an initial SYN segment (*id.*, 18:66–67) to the client to negotiate the connection. The client can then choose to “accept these parameters ... or propose different parameters” in its responsive SYN message. (*Id.*, 24:43–45.) Thus, the client necessarily receives null segment timeout value (“idle information”), one of the parameters, upon receiving the SYN segment from the server. (EX1002 ¶¶206–208; EX1007, 18:58–61; *see also supra* Section IX.A.1.d.)

To the extent the *Wookey-Berg* combination does not disclose the claimed “receiving,” it would have been obvious to modify the functionality in the *Wookey-Berg* system/process such that it receives the negotiable parameters. (EX1002 ¶209.) A POSITA would have been motivated to configure the *Wookey-Berg* system/process to include functionality for receiving the parameters, including null segment timeout value (“idle information”), to facilitate the subsequent process of (i) including the client’s desired parameters for the RUDP connection in the SYN segment or (ii) accepting or rejecting the parameter in the SYN packet consistent with the features described by *Berg*. (EX1007, 24:44–48.) Therefore, the above-discussed *Wookey-Berg* system/process discloses and/or suggests such features to the extent not disclosed. (EX1002 ¶¶209–210.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

Accordingly, for reasons similar to those explained above, a POSITA would have been motivated to implement in the *Wookey-Berg* combined process of establishing the connection between **remote machine 30'** and **client machine 10**, the process of receiving, by **client machine 10**, the null segment timeout value (“idle information”) for use in detecting an idle time period that results in an RUDP (“non-TCP”) connection being subject to deactivation. (*See supra* Section IX.A.1.b; EX1002 ¶¶211–212.)

- d) **generating, based on the idle information, a non-TCP packet including an idle time period parameter field identifying metadata that is specified in at least one of a number of seconds or minutes; and**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶213–220.) For example, in the combined The *Wookey-Berg* system/method, **client machine 10** generates, based on the identified null segment timeout value (“idle information”), an RUDP SYN segment (“non-TCP packet”) including the timeout value, specified in seconds, in an associated parameter field in two ways. (EX1002 ¶213; *supra* Sections IX.A.1.b–e.)

As explained above, *Berg* discloses generating an RUDP SYN segment (“non-TCP packet”) for establishing an RUDP connection. (*Supra* Section IX.A.1.c.) The RUDP connection is established by the exchange of RUDP SYN segments, which includes parameter fields for all of the proposed negotiable parameters for the RUDP

Petition for *Inter Partes* Review
Patent No. 10,306,026

connection, between the two hosts. (*Id.*; EX1007, 18:57–61, 24:40–47; 24:40–48.) Further, as explained for limitation 52.d, the SYN segment includes a field (“idle time period parameter field”) for identifying the null segment timeout value (“metadata”) that is specified in milliseconds, a number of seconds, by the ULP/application. (*Supra* Section IX.A.1.d; EX1007, 20:36–42, 20:43–48, 24:40–42, 19:1–40; EX1002 ¶¶214–215.)

First, when the client generates the initial RUDP SYN segment, the ULP/application of the client identifies the negotiable parameters, including the identified null segment timeout value (“idle information”), for the SYN segment via an API. (EX1007, 24:40–48; *see also id.*, 18:57–61; *supra* Section IX.A.6.c, IX.A.1.d.) The client then “initiates a connection” by sending the generated “SYN segment which contains the negotiable parameters” to the server. (EX1007, 24:40–47; EX1002 ¶216.)

Second, when the client receives the initial SYN segment from the server, the client can then choose to “accept these parameters by echoing them back in its SYN message or propose different parameters in its SYN response,” i.e., the generated responsive SYN segment is based on the claimed idle information (e.g., the identified null segment timeout value). (EX1007, 24:43–45; EX1002 ¶217; *supra* Sections IX.A.6.c, IX.A.1.d.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been further configured to include generating, by **client machine 10**, based on the null segment timeout value (“idle information”), an RUDP SYN segment (“non-TCP packet”) including a parameter field (“idle time period parameter field”) identifying the null segment timeout value (“metadata”) that is specified in a number of seconds or minutes. (*See supra* Section IX.A.1.b.)

Thus, the *Wookey-Berg* combination discloses and/or suggests limitation 61.d. (EX1002 ¶¶218–220.)

- e) **sending, from the node to another node and for establishing the non-TCP connection, the non-TCP packet to provide the metadata to the another node, for use by the another node in determining a timeout attribute associated with the non-TCP connection;**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶221–226.) For example, **client machine 10** (“node”) sends an RUDP SYN segment (“the non-TCP packet”) to **remote machine 30’** (“another node”) to provide the negotiable parameters, including the null segment timeout value (“metadata”), to establish the RUDP (“non-TCP”) connection. (EX1002 ¶221.) **Remote machine 30’** uses the received metadata to determine a timeout attribute associated with the RUDP connection, as explained below. (*Id.*)

Petition for *Inter Partes* Review
Patent No. 10,306,026

First, when the client initiates an RUDP connection, the client sends the initial RUDP SYN segment “which contains the negotiable parameters” including the identified null segment timeout value (“metadata”) to the server. (EX1007, 24:40–47.) Second, when the client receives the initial SYN segment from the server, the client can then choose to “accept these parameters by echoing them back in its SYN message or propose different parameters in its SYN response.” (*Id.*, 24:43–45; EX1002 ¶222.)

In either scenario, the server enables and maintains its own null segment timer (“timeout attribute”) set to a duration/timeout value twice that of the client’s null segment timeout value (received “metadata” from the SYN message). (EX1007, 23:45–52.) When the server’s null segment timer expires, an “auto reset is initiated.” (EX1007, 23:45–57; EX1002 ¶¶223; *supra* Section IX.A.1.d.)

Thus, the null segment timeout value (provided “metadata”) is used to determine the attributes of the server’s null segment timer (“timeout attribute”) associated with the RUDP connection. (EX1002 ¶224.)

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been further configured to include the process of sending, from **client machine 10** to **remote machine 30**, and for establishing the RUDP connection, the RUDP SYN segment to provide the null segment timeout

value (“metadata”) to **remote machine 30’**, for use by **remote machine 30’** in determining a null segment timer (“timeout attribute”) associated with the RUDP connection. (*See supra* Section IX.A.1.b; EX1002 ¶¶225–226.)

- f) **wherein a three-way TCP handshake is performed for establishing a TCP connection that is separate from the non-TCP connection.**

The *Wookey-Berg* combination discloses and/or suggests this limitation for at least the same reasons as above for limitations 52.f. (*See supra* Section IX.A.1.f; EX1002 ¶¶227–228.)

7. Claim 64

- a) **The method of claim 61 wherein: one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period;**

The *Wookey-Berg* combination discloses and/or suggests this limitation for at least the same reasons above for claim 55. (*Supra* Section IX.A.2.a; EX1002 ¶229.)

- b) **the keep-alive period is not configurable, while the idle time period is configurable; and**

The *Wookey-Berg* combination discloses and/or suggests this limitation for at least the same reasons above for claim 56. (*Supra* Section IX.A.3.a; EX1002 ¶230.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

- c) **the one or more keep-alive packets are conditionally communicated based on at least one keep-alive attribute.**

The *Wookey-Berg* combination discloses and/or suggests this limitation for at least the same reasons above for claim 57. (*Supra* Section IX.A.4.a; EX1002 ¶231.)

8. Claim 66

- a) **The method of claim 64 and further comprising: configuring the node to cause the another node to determine, based on the metadata, the timeout attribute associated with the non-TCP connection, in connection with negotiating, with the node, a duration of the idle time period.**

The *Wookey-Berg* combination discloses this claim for at least the same reasons as above for claims 59 and 64. (*See supra* Sections IX.A.5, IX.A.7.a–c; EX1002 ¶232.)

9. Claim 68

- a) **A method comprising:**

To the extent limiting, *Wookey* discloses this limitation for at least the same reasons as above for claim element 52.a. (*See supra* Sections IX.A.1.a; EX1002 ¶233; *infra* Sections IX.A.8.b–g.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

- b) **at a node configured with a network application that is configured to operate in accordance with a first protocol including a transmission control protocol (TCP) to establish a TCP connection:**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for at least the same reasons discussed above for limitations 52.b and 52.f. (*Supra* Sections IX.A.1.b.1, IX.A.1.f; EX1002 ¶¶234–235.) For example, **client machine 10** (“node”) in the *Wookey-Berg* combination is configured with a web browser application 280 (“network application”) that is configured to operate in accordance with a TCP to establish a TCP connection with **remote machine 30**. (*Supra* Sections IX.A.1.b, IX.A.1.f; EX1002 ¶235.)

- c) **based on operating in accordance with a second protocol, that is separate from the TCP, to establish a second protocol connection:**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for at least the same reasons discussed above for limitations 52.b and 52.f. (*Supra* Section IX.A.1.b, IX.A.1.f, EX1002 ¶¶236–238.) For example, **client machine 10** in the combined *Wookey-Berg* system/method would utilize the web browser application 280 to operate per the RUDP protocol (“a second protocol”), which is separate from the TCP protocol, to establish a RUDP connection with **remote machine 30**. (*Supra* Sections IX.A.1.b, IX.A.1.f; EX1001, 32:51 (“the second protocol includes a non-TCP protocol”).)

Petition for *Inter Partes* Review
Patent No. 10,306,026

d) **receiving, from another node, a packet;**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for at least the same reasons discussed above for limitations 52.c. (*Supra* Section IX.A.1.c, EX1002 ¶¶239–240.) For example, **client machine 10** in the combined *Wookey-Berg* system/method receives, from **remote machine 30'** (“another node”), an RUDP SYN message (“packet”). (*Supra* Section IX.A.1.c.)

e) **detecting an idle time period parameter field in the packet;**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for at least the same reasons discussed above for limitations 52.d. (*Supra* Section IX.A.1.d, EX1002 ¶¶241–242.) For example, **client machine 10** in the combined *Wookey-Berg* system/method would detect the parameter field (“an idle time period parameter field”) that contains the null segment timeout value in the received RUDP SYN message (“packet”). (*Supra* Section IX.A.1.d; EX1002 ¶¶241.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

- f) **identifying metadata in the idle time period parameter field for an idle time period, where, in response to the idle time period being detected, the second protocol connection is deemed inactive; and**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation. (EX1002 ¶¶243–243.) As discussed above for limitation 52.d and further below, *Berg*’s RUDP protocol includes processes for identifying a null segment timeout value (“metadata”) in the parameter field (“idle time period parameter field”) of a RUDP SYN segment. (*Supra* Section IX.A.1.d.) Further, the identified null segment timeout value (“metadata”) is for detecting an idle time period, where, in response to the idle time period being detected, the RUDP connection (“second protocol connection”) is deemed inactive. (EX1002 ¶243.)

For example, *Berg* explains that the purpose of the null segment mechanism is to determine whether the “connection is *still active*” by sending a null segment to the other node. (EX1007, 22:25–32.) The null segment mechanism in *Berg* utilizes a timer to control when to send a null segment. (*Id.*, 23:45–49.) When the client’s null segment timer expires, it sends a null segment to the server. (*Id.*, 23:45–49.) The client’s null segment timer’s duration is set to the null segment timeout value (“metadata”). (*Id.*) Therefore, the null segment timeout value (“metadata”) defines a duration of time that the connection will remain inactive before the client sends a null segment to see if the connection still exists. (*Id.*, 20:36–41, 22:25–32; EX1002

Petition for *Inter Partes* Review
Patent No. 10,306,026

¶244.) Accordingly, a POSITA would have therefore understood that the identified null segment timeout value (“metadata”) is for detecting an idle time period, where, in response to the null segment timer expiring (“idle time period being detected”), the RUDP connection (“second protocol connection”) is deemed inactive. (EX1002 ¶244.)

The above understanding is consistent with how the ’026 patent describes its “detecting” of an idle time period based on detecting a timer expiration. (EX1001, 18:33–34; EX1002 ¶245.)

Accordingly, for reasons similar to those explained above, the *Wookey-Berg* combined process would have been further configured to include **client machine 10** identifying the null segment timeout value (“metadata”) in the null segment timeout value parameter field (“idle time period parameter field”) in the RUDP SYN segment (“non-TCP packet”), for an idle time period, where, in response to the idle time period being detected, the RUDP (“second protocol”) connection is deemed inactive, as claimed. (EX1002 ¶¶246–247; *see supra* Section IX.A.1.b.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

- g) **creating or modifying, based on the metadata, a timeout attribute associated with the second protocol connection.**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for at least the same reasons discussed above for limitations 52.e and claim 55. (*Supra* Sections IX.A.1.e, IX.A.2; EX1002 ¶¶248–249.) For example, as discussed above, the **client machine 10** in the *Wookey-Berg* combination may enable or disable (e.g., “creat[es]”/“modif[ies]”) the null segment mechanism (a “timeout attribute”), based on the negotiated null segment timeout value (“metadata”), associated with the RUDP (“second protocol”) connection. (*Supra* Sections IX.A.1.e, IX.A.2; EX1002 ¶248.)

10. Claim 69

- a) **The method of claim 68 wherein creating or modifying the timeout attribute renders one or more keep-alive packets in the second protocol connection unnecessary.**

The *Wookey-Berg* combination discloses and/or suggests and renders obvious this limitation for at least the same reasons discussed above for claims 57 and 68. (*Supra* Sections IX.A.2–5, IX.A.9; EX1002 ¶¶250–251.) As discussed above, the **client machine 10** in the *Wookey-Berg* combination may disable (“creat[es]”/“modif[ies]”) the null segment mechanism (“timeout attribute”). (*Supra* Sections IX.A.2–5.) A POSITA would have understood that disabling the null segment mechanism for the connection indicates to the nodes that one or more null segments (“keep-alive packets”) in the RUDP (“second protocol”) connection are unnecessary. (EX1002 ¶250.)

B. Ground 2: The Combination of *Wookey*, *Berg*, and *Tucker* Render Obvious Claims 60 and 67**1. Claim 60**

- a) **The method of claim 55 wherein a duration of the idle time period is configured by an application via an analog of a socket.**

Wookey in view of *Berg* and *Tucker* discloses and/or suggests and renders obvious the limitations of claim 60. (EX1002 ¶¶253–272.) As discussed above for claims 52 and 55, the **client machine 10** in the *Wookey-Berg* combination initiates the process for establishing a RUDP connection with **remote machine 30'** that includes the negotiation of a null segment timeout value (“a duration of the idle time period”). (*Supra* Sections IX.A.2; IX.A.1.b–f.)

Berg further discloses that when the **client** initiates a connection, the client “sends a SYN segment which contains the negotiable parameters *defined by the ULP via the API.*” (EX1007, 24:39–48.) A POSITA would have thus recognized that the null segment timeout is configured—e.g., when proposed and negotiated—by a **client**’s application via an interface operating between the application and the RUDP protocol. (EX1002 ¶254; EX1007, 24:48–67.)

An inter-process communication (“IPC”) would be used to provide the interface between the application/user and the RUDP protocol. (EX1002 ¶255; *see also infra* Section IX.B.1.a.1 (explaining that a socket is one example of an IPC

Petition for *Inter Partes* Review
Patent No. 10,306,026

implementation).) As discussed, *Berg*'s RUDP protocol includes functionality for negotiating a null segment timeout value. (EX1007, 24:39–48.) The RUDP protocol includes an API interface between the protocol and the application/user (*id.*, 24:48–67) to allow the exchange of data between the protocol and the application/user—e.g., an interface for facilitating data exchange related to proposing/negotiating a null segment timeout value (*id.*, 24:48–67). (EX1002 ¶255.)

This interface would have allowed requests for exchanging data between the user programs (e.g., application) and the RUDP protocol. (*Id.* ¶256; EX1007, 24:48–67.) Thus, a POSITA would have understood that *Berg*'s API provides the basic functions that the RUDP must perform to support interprocess communication between the application and the RUDP protocol to configure connection parameters, such as the null segment timeout value. (EX1002 ¶256.)

While *Berg*'s RUDP protocol, as implemented in the *Wookey-Berg* combination, provides an API interface between the application and the RUDP protocol (at the **client machine 10** in the *Wookey-Berg* combination) to facilitate the configuration of an null segment timeout value for negotiation with **remote machine 30'**, *Berg* does not expressly provide the specific IPC mechanism used to implement the interface. For example, *Berg* explains that the “details of the API are dependent on the platform.” (EX1007, 24:65–67.) Nevertheless, it would have been obvious

Petition for *Inter Partes* Review
Patent No. 10,306,026

to implement *Berg*'s API, as implemented in the *Wookey-Berg* combination, between the protocol and the application/user with an "analog of a socket" to allow the exchange of a timeout value between the application and protocol based on the teachings/suggestions of *Tucker* and a POSITA's knowledge. (EX1002 ¶257.)

(1) Tucker

Tucker, like *Berg*, relates to the communication of messages and data between different processes. (EX1002 ¶¶258, 98–99 (*Tucker* overview); compare EX1031, Abstract with EX1007, 24:48–67.) *Tucker* introduces common IPCs typically found in an operating system (OS), such as "a shared memory segment, a message queue, . . . , a pipe, a stream, a socket" (EX1031, 3:36–43), which are utilized by an operating system to allow communication (e.g., the exchange messages or commands) between different processes or programs. (See also EX1031, 2:36–39, 3:11–15, 3:45–50, 8:52–56, 9:37–45, 10:24–31 (explaining that there are a variety of mechanisms available for providing IPC capability), FIG. 2A; EX1002 ¶259.)

A POSITA would have understood that a designer may choose from a variety of IPC objects when enabling IPC capability between processes (e.g., an application and a RUDP protocol as discussed above) to meet different system objectives, such as performance, modularity, and system circumstances. (EX1002 ¶260; EX1031, 10:24–26.) Accordingly, to enable communications between processes, a designer

Petition for *Inter Partes* Review
Patent No. 10,306,026

may select an available IPC mechanism in an OS environment. (*E.g.*, EX1031, 8:59–61, 10:54–63, FIG. 2A; EX1002 ¶260.)

For example, a socket allows data communication to be sent over a network interface, either to a different process on the same machine or to another machine on the network. (EX1002 ¶261; *see also* EX1031, 13:16–22, 14:50–56.) *Tucker* explains that a socket can be interchanged with other IPC components such as a shared memory segment, a message queue, a pipe, or a stream for establishing an IPC. (EX1031, 13:55–65.) These non-socket IPC components are similar to a socket component in that they each facilitate communication between processes on a machine. (EX1002 ¶261.) Thus, a shared memory segment, a message queue, a pipe, and a stream are each an “analog of a socket” component for providing inter-process communication. (EX1002 ¶261; *see also id.* ¶262 (discussing other IPC components in *Tucker* that are also analogs of a socket).)

Tucker’s disclosures regarding analogs of a socket are consistent with the teachings of the ’026 patent. For example, the ’026 patent explains that the socket—and the analog of a socket component—allows data exchange between the network application layer and the TCP layer. (*E.g.*, EX1001, 11:2–4, 12:11–13, 12:23–25.) While the ’026 patent does not provide an explicit definition of what an analog of a socket is, the ’026 patent discloses other communication mechanisms within a

Petition for *Inter Partes* Review
Patent No. 10,306,026

system, including “a message queue,” “pipe,” or “shared location in IPU memory and/or secondary storage.” (*Id.*, 11:65–12:3; EX1002 ¶263.)

Moreover, the understanding that *Tucker*’s descriptions regarding a stream teaching an analog of a socket is consistent with PO’s view of the same claimed features. (EX1017, 20–22 (mapping an “analog of a socket” to a “stream socket”), 82–84 (same).)²²

(2) Reasons to Combine

Based on the teachings of *Tucker* and a POSITA’ knowledge at the time, it would have been obvious to configure and implement the interface between the RUDP protocol and the application/user at the **client machine 10** in the *Wookey-Berg* combination to use an IPC component, such as a shared memory segment, a message queue, a pipe, or a stream (each disclosing “an analog of a socket”) to enable the application to configure the null segment timeout value (“the duration of the idle time period”) for the RUDP connection between the application and the RUDP protocol. (EX1002 ¶¶264–272.) Such an implementation would have had

²² Petitioner’s references to PO’s infringement allegations are not indicative of any concession that any accused instrumentality infringes any claim limitation as alleged by PO.

Petition for *Inter Partes* Review
Patent No. 10,306,026

many advantages and been a predictable and straightforward result achieved by combining well-known technologies using known methods. *See KSR*, 550 U.S. at 416–18.

Tucker and *Berg* disclose features in a similar technological field. (EX1002 ¶265; *supra* Section IX.B.1.a.1.) Like *Berg*, *Tucker* discloses features for allowing the communication of data and commands between different processes (e.g., EX1031, 10:24–31), so a POSITA would have had reason to consider *Tucker* when contemplating and implementing the teachings of *Tucker* in the *Wookey-Berg* combination.

A POSITA would have been motivated to incorporate *Tucker*’s teachings and suggestions of creating and using an IPC component, such as the above-discussed non-socket IPC components, for *Berg*’s API interface (e.g., at the **client machine 10** in the *Wookey-Berg* combination) to facilitate the configuration of a null segment timeout value between the application and the RUDP protocol for negotiation of a null segment timeout value with **remote machine 30**’. (EX1002 ¶266.) Using any one of these non-socket IPC components as described in *Tucker* in the combined *Wookey-Berg*’s system/process as the mechanism to facilitate exchange of parameters, such as the null segment timeout value between the application and the RUDP protocol, would have achieved the required communication between the

Petition for *Inter Partes* Review
Patent No. 10,306,026

application and the RUDP protocol (e.g., at the **client machine 10** in the *Wookey-Berg* combination). (*Id.*) Indeed, IPC implementations, like those disclosed in *Tucker*, are applicable in a wide variety of networking environments including both TCP and UDP based environments. (*Id.* (citing EX1033–EX1034).) Furthermore, the above-modification would have involved the substitution of one known IPC component (e.g., a socket) with another known IPC component (e.g., a shared memory segment, a message queue, a pipe, or a stream) from a finite number of available IPC components such as those described by *Tucker*. (*Id.* ¶267.)

Given that the **client machine 10**'s application in the *Wookey-Berg* combination must communicate with the RUDP protocol as discussed above, configuring the *Wookey-Berg* combination to utilize the above-noted non-socket IPC components for the intercommunication between the application and the RUDP protocol based on the teachings of *Tucker* would have been both a predictable and straightforward implementation. (*Id.* ¶268); *see KSR*, 550 U.S. at 416. A POSITA would have had the skills and knowledge to achieve this implementation by combining well-known technologies using known methods (such as the above-discussed non-socket IPC components, which are common mechanisms in operating systems), and technologies as described above by *Tucker* and *Berg*, and known in the art at the time. (EX1002 ¶268.) Accordingly, a POSITA would have understood

Petition for *Inter Partes* Review
Patent No. 10,306,026

that such a combination would have yielded the predictable result of providing an IPC between the application and the RUDP protocol. (*Id.* ¶269); see *KSR*, 550 U.S. at 417.

A POSITA would have also recognized that the *Wookey-Berg-Tucker* combination would have involved using known technologies (e.g., IPC components) and design concepts and processes to obtain the foreseeable result at a node of an IPC between the application and the RUDP protocol using the above-discussed non-socket IPC components. (EX1002 ¶270.) Thus, a POSITA would have known how to configure the *Wookey-Berg* method to implement *Tucker*'s above-discussed non-socket IPC component implementations, and would have had a reasonable expectation of success in the above-modification. (*Id.* ¶¶270–271.) *Pfizer*, 480 F.3d at 1364.

* * *

Thus, the *Wookey-Berg-Tucker* combination renders obvious claim 31. (EX1002 ¶272.)

2. Claim 67

- a) **The method of claim 64 wherein a duration of the idle time period is configured by an application via an analog of a socket.**

The *Wookey-Berg-Tucker* combination discloses this claim for at least the same reasons as above for claims 60 and 64. (*See supra* Sections IX.A.7.a–c, IX.B.1; EX1002 ¶¶273–274.) For example, a duration of the idle time period (“null segment timeout value”) is configured by **client machine 10**’s application via an analog of a socket (such as a shared memory segment, a message queue, a pipe, or a stream). (*Supra* Section IX.B.1.)

X. DISCRETIONARY DENIAL IS NOT APPROPRIATE

The Board should not exercise its discretion to deny institution under 35 U.S.C. § 314(a). *NHK Spring Co., Ltd. v. Intri-Plex Techs, Inc.*, IPR2018-00752, Paper 8 (Sept. 12, 2018) does not apply here as no trial date has been scheduled in the related litigation. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 3 (Mar. 20, 2020) (precedential). The six-factor test addressed in *Fintiv* (“*Fintiv* factor(s)”) favors institution. *See id.*, 5–6.

The **first *Fintiv* factor** (stay) is neutral, if not weighing in favor of institution. While there is currently no stay pending IPR resolution before the Western District of Texas (“WDTX”) court, Google intends to pursue such a stay should institution be granted. The Board should not speculate as to how the court would rule on the requested stay based on actions taken in different cases. *Western Digital Corp. et al. v. Martin Kuster*, IPR2020-01391, Paper 10 at 8–9 (February 16, 2020) (finding this factor neutral despite Patent Owner assertions regarding WDTX Judge Albright’s past practices regarding stays). Moreover, Google filed a motion to transfer in the district court litigation. The Court has currently stayed the litigation pending resolution of Google’s transfer motion. (EX1030.)

The **second *Fintiv* factor** (proximity of trial dates) weighs strongly in favor of institution. *First*, the district court has not set a trial date, which “weighs

Petition for *Inter Partes* Review
Patent No. 10,306,026

significantly against exercising [] discretion to deny institution of the proceeding.” *Google LLC v. Uniloc 2017 LLC*, IPR2020-00441, Paper 13 at 35 (July 17, 2020); *Nested Bean, Inc. v. Big Beings USA PTY LTD*, IPR2020-01234, Paper 15 at 14 (Jan. 25, 2021) (same). In fact, while the parties submitted proposed schedules, the court granted Google’s motion to stay the case pending resolution of Google’s transfer motion and declined to set a case schedule.²³ (EX1030; EX1024.) Accordingly, determining at this time whether the Board’s anticipated FWD date (e.g., September–October 2022) will occur after trial in the related litigation would be speculative.

Even if the parties’ most recent proposed schedule was considered as a baseline for comparison (which Google submits would be improper at this time), dates in that proposed schedule would need to be delayed because they were proposed prior to the court’s recent stay order and did not take the stay into consideration. (EX1030; EX1024, 5–9.) Accordingly, the proposed dates in light

²³ Disposition of Google’s transfer motion has taken priority over other activities. (EX1035.) *See also In re Apple Inc.*, 979 F.3d 1332, 1337 (Fed. Cir. 2020) (“disposing of a filed transfer motion ‘should unquestionably take top priority.’”).

Petition for *Inter Partes* Review
Patent No. 10,306,026

of the court’s stay order demonstrates that trial will likely occur after August 2022.²⁴ (EX1024, 6, 9.) Thus, any projected trial date (*albeit* speculative) would be more likely after the expected due date of the Board’s FWD (e.g., around September-October 2022).

The Board has instituted trial even where an *actual* litigation trial date precedes the FWD deadline by two months or more. *Sand Revolution II, LLC v. Cont’l Intermodal Grp–Trucking LLC*, IPR2019-01393, Paper 24 at 8–10 (June 16, 2020) (“*Sand*”) (informative) (factor 2 favored institution despite WDTX trial preceding FWD deadline by five months); *Western Digital Corp.*, IPR2020-01391 at 9-10; *Apple Inc. v. Parus Holdings, Inc.*, IPR2020-00686, Paper 9 at 11–13, 22; *Fintiv*, IPR2020-00019, Paper 11 at 5, 9.

It is also unlikely that trial in WDTX will proceed without delay. WDTX has issued a suspension order every month for the past twelve months suspending almost

²⁴ Consistent with the court’s practice, the proposed order stages deadlines and trial date based on the proposed August 2021 *Markman* hearing and that the “actual trial date” may materially differ from the schedule and that the court would consider “reasonable amendments to the case schedule post-*Markman*.” (EX1024, 9 n.3; *id.*, 6.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

all trials in the district from March 13, 2020 to at least April 30, 2021 due to the COVID-19 pandemic, creating a large backlog of trials. (EX1025; *see also* EX1026, 2 (agreeing to stipulation to postpone trial date “to allow the COVID-19 situation to ameliorate”); EX1029 (showing J. Albright’s docket including 792 cases filed in 2020).) Thus, as the Board has recognized, civil trials in WDTX “may possibly slip ... [due to] months of backlogged trials, including many active criminal cases that would take precedence over civil trials.” *HP Inc. v. Slingshot Printing LLC*, IPR2020-01085, Paper 12 at 7 (Jan. 14, 2021); *Sand* at 8-10.

Finally, the potential for transfer based on Google’s transfer motion further reduces the likelihood that trial will precede the Board’s FWD. *Dish Network, L.L.C. v. Broadband iTV, Inc.*, IPR2020-01267, Paper 15 at 17–18 (Jan. 21, 2021) (explaining that the second *Fintiv* factor “is a proxy for the *likelihood* that the trial court will reach a decision on validity issues before the Board reaches a [FWD]”). Indeed, transfer of the WDTX case to the proper venue (the Northern District of California) will undoubtedly delay any trial beyond the anticipated FWD due date. *C.f. Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 8, at 7 n.7 (Mar. 27, 2020) (denying institution based in part on scheduled trial date that was later vacated upon transfer); *see Uniloc 2017 LLC v. Google LLC*, Case No. 18-CV-00502, Dkt No. 277 (E.D. Tex. June 19, 2020) (transfer order vacating unreached deadlines).

Petition for *Inter Partes* Review
Patent No. 10,306,026

The **third *Fintiv* factor** (investment in parallel proceedings) also weighs in favor of institution. The case is in earliest stages, and as discussed, all deadlines other than those concerning Google’s transfer motion have been stayed. Thus, investment by the parties in invalidity and claim constructions issues will be delayed, especially if the case is transferred. And even if the transfer motion is denied, using the above-discussed proposed schedule as an exemplary baseline (which will require adjustments), any *Markman* hearing in the case is likely not to occur until August 2021 or later, and thus would occur near the expected due date of the Board’s institution decision. (EX1024, 6.) Discovery will not commence, and dispositive motions will not be due, until after the *Markman* hearing. Other case activity that has taken place or will eventually take place (outside claim construction) that does not relate to validity should “not weigh in [the Board’s] consideration of this issue.” *Western Digital Corp.*, IPR2020-01391 at 11. Such minimal investment is insufficient to support discretionary denial. *Juniper Networks, Inc. v. Huawei Digital Techs. (Cheng Du) Co., Ltd.*, IPR2020-01130, Paper 13 at 12–13 (Jan. 22, 2021) (factor 3 weighs against discretionary denial when claim construction order has been entered, but fact and expert discovery has not closed, and dispositive motions and briefings have not been submitted); *Dish* at 19–21 (similar); *HP* at 7 (similar). Additionally, Google’s diligence in filing this Petition just four months

Petition for *Inter Partes* Review
Patent No. 10,306,026

after receiving PO's narrowed list of asserted claims²⁵ further weighs against discretionary denial. *Dish* at 20–21 (petitioner's diligence in filing the petition weighed against discretionary denial); *Fintiv* at 11.

The **fourth *Fintiv* factor** (overlap) is neutral if not in favor of institution. There is, at this early stage of the district court litigation, no evidence of overlap with the grounds here and any invalidity positions to be pursued in the district court. Indeed, preliminary invalidity contentions have not yet been served. And, as discussed above, the district court must first resolve the pending transfer motion before addressing substantive issues, including invalidity. Given the stay of the litigation, it is likely expert discovery relating to invalidity will not be completed until at the earliest mid-2022 and dispositive motions on those issues would be due

²⁵ PO initially asserted 455 claims across eight patents (including 107 claims from the '026 patent). (EX1032, 1.) The WDTX court ordered PO to substantially reduce the asserted claims to approximately 65 claims. On October 20, 2020, PO narrowed its asserted claims as ordered, but “reserve[d] the right to either narrow the claims further, **substitute claims**, or **add a reasonable number of claims**.” (EX1027.) On December 4, 2020, PO informed Petitioner that it will not modify its narrowed list of asserted claims “**at this time**.” (EX1028.)

Petition for *Inter Partes* Review
Patent No. 10,306,026

after that. Accordingly, substantive invalidity issues would not be addressed in the litigation until well past the institution decision and likely very near the time of the Board's expected FWD. Moreover, Petitioner may rely on invalidity grounds not asserted here in the district court litigation, including system art and prior art references and combinations not raised herein. Therefore, any potential overlap between this proceeding and the district court is minimal and speculative at this time.

Other circumstances (***Fintiv* factor six**) also favor institution. The Petition presents strong grounds demonstrating the unpatentability of the challenged claims, which weighs against discretionary denial. (*See supra* Section IX.) *Western Digital Corp.*, IPR2020-01391 at 14-15. Indeed, the Board has already instituted the Google -845 IPR and the Unified -742 IPR challenging claims of the related '995 patent. (*See supra* Section II.) The Unified -742 IPR is based in part on *Berg*, which is being applied here. Moreover, this petition and the concurrently-filed sister petition (*id.*) are the only challenges to the '026 patent before the Board, which is a "crucial fact" favoring institution. *Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 10 at 6 (May 12, 2020).²⁶

²⁶ Discretionary denial under *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper No. 19 is not applicable here given the

Petition for *Inter Partes* Review
Patent No. 10,306,026

While ***Fintiv* factor 5** (same parties) may weigh slightly in favor of denial, the remaining factors are at least neutral, if not strongly favoring institution. Further, even if the Board finds that **factor 1** (or even **factor 2**) somehow tilts in favor of denial, Petitioner’s diligence, the lack of relevant investment of resources, lack of evidence of overlap, and/or strength of Petitioner’s grounds (**factors 3, 4, 6**) outweigh these other factors. *See e.g., SK Hynix Inc. et al. v. Netlist, Inc.*, IPR2020-01421, Paper 10 at 6–13 (Mar. 16, 2021). Accordingly, based on a “holistic view” of whether integrity of the system and efficiency is best served, institution here is proper. *Samsung Elecs. Co. Ltd. v. Dynamics Inc.*, IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

’026 patent is not at issue in any other proceeding pending before the Board (outside of Petitioner’s concurrently-filed sister petition).

Petition for *Inter Partes* Review
Patent No. 10,306,026

XI. CONCLUSION

For the reasons provided, Petitioner requests institution of IPR for claims 52, 55–57, 59–61, 64, 66–69 of the '026 patent based on the ground specified in this petition.

Respectfully submitted,

Dated: April 29, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,306,026

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,306,026 contains, as measured by the word-processing system used to prepare this paper, 13,991 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: April 29, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

Petition for *Inter Partes* Review
Patent No. 10,306,026

CERTIFICATE OF SERVICE

I hereby certify that on April 29, 2021, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 10,306,026 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on PAIR:

Patrick Caldwell (Reg. No. 44580)
G. Gordon (Reg. No. 64517)
P.O. Box 59655
Dept. SVIPGP
Dallas, TX 75229

A courtesy copy was also sent via electronic mail to the Patent Owner's litigation counsel at the following addresses:

TD-PTAB@devlinlawfirm.com
ddahlgren@devlinlawfirm.com
dlflitparas@devlinlawfirm.com

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)

EXHIBIT 11

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

JENAM TECH, LLC,
Patent Owner

Patent No. 10,742,774

**PETITION FOR POST GRANT REVIEW
OF U.S. PATENT NO. 10,742,774**

Petition for Post Grant Review
Patent No. 10,742,774

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES UNDER 37 C.F.R. § 42.8.....	1
A.	Real Parties-in-Interest.....	1
B.	Related Matters.....	1
1.	Related Lawsuit.....	1
2.	Related Applications	1
3.	Related IPRs.....	2
C.	Counsel and Service Information.....	3
III.	PAYMENT OF FEES UNDER 37 C.F.R. § 42.15(a)	4
IV.	TIME FOR FILING UNDER 37 C.F.R. § 42.202	4
V.	GROUND FOR STANDING UNDER 37 C.F.R. § 42.204(a).....	4
VI.	PRECISE RELIEF REQUESTED AND GROUNDS RAISED.....	4
A.	Claim for Which Review Is Requested.....	4
B.	Statutory Grounds of Challenge.....	4
VII.	LEVEL OF ORDINARY SKILL IN THE ART.....	6
VIII.	OVERVIEW OF THE '774 PATENT	7
IX.	THE '774 PATENT IS ELIGIBLE FOR PGR.....	10
A.	The Pre-AIA Applications Do Not Support the “Non-TCP” Features Recited in Claim 42	13
1.	The Pre-AIA '454 Application Does Not Support the “Non-TCP” Features	15
2.	The Pre-AIA '402 Application Does Not Support the “Non-TCP” Features	21

Petition for Post Grant Review
Patent No. 10,742,774

3.	Materials Incorporated by Reference in the Two Pre-AIA Applications Do Not Support the “Non-TCP” Features.....	22
B.	AIA Applicability.....	26
X.	CLAIM CONSTRUCTION	26
XI.	DETAILED EXPLANATION OF GROUNDS.....	28
A.	Ground 1: Claim 1 Is Indefinite	28
1.	The Claim Language, Specification, and File History Do Not Sufficiently Describe the “TCP-Variant” Features.....	30
2.	The Intrinsic Record Does Not Inform a POSITA About the Scope of the Claimed “TCP-Variant” Features Recited With Reasonable Certainty	34
B.	Ground 2: The Combination of <i>Wookey</i> and <i>Eggert</i> Renders Obvious Claim 1	48
1.	Claim 1	48
XII.	<i>EGGERT</i> IS A PRINTED PUBLICATION.....	88
XIII.	DISCRETIONARY DENIAL IS NOT APPROPRIATE	89
A.	35 U.S.C. § 325(d) Does Not Provide a Basis for Discretionary Denial	89
B.	The Related Litigation Provides No Basis for Discretionary Denial	92
XIV.	CONCLUSION.....	100

Petition for Post Grant Review
Patent No. 10,742,774

LIST OF EXHIBITS

EX1001	U.S. Patent No. 10,742,774
EX1002	Declaration of Bill Lin, Ph.D.
EX1003	Curriculum Vitae of Bill Lin, Ph.D.
EX1004	File History of U.S. Patent Application No. 16/368,811 (U.S. Patent No. 10,742,774)
EX1005	U.S. Pre-Grant Publication No. 2007/0171921 to Wookey <i>et al.</i>
EX1006	L. Eggert, TCP Abort Timeout Option, draft-eggert-tcp- tcp-abort-timeout-option-00, Network Working Group, Internet-Draft (April 14, 2004) (“ <i>Eggert</i> ”)
EX1007	U.S. Patent No. 6,981,048 to Abdolbaghian
EX1008	DARPA RFC 793 TRANSMISSION CONTROL PROTOCOL
EX1009	IETF RFC 1122 “Requirements for Internet Hosts -- Communication Layers”
EX1010	U.S. Patent No. 9,923,995
EX1011	IETF RFC 1001 “Protocol Standard for a NetBIOS Service On A TCP/UDP Transport: Concepts and Methods”
EX1012	U.S. Patent No. 6,212,175 to Harsch
EX1013	U.S. Patent No. 6,665,727 to Hayden
EX1014	U.S. Patent No. 7,636,805 to Rosenberg
EX1015	IETF RFC 5440 “Path Computation Element (PCE) Communication Protocol (PCEP)”
EX1016	IETF RFC 5482 “TCP User Timeout Option”

Petition for Post Grant Review
Patent No. 10,742,774

EX1017	<i>Jenam Tech, LLC's Second Amended Set of Infringement Contentions</i> regarding U.S. Patent No. 10,742,774 (March 17, 2021)
EX10181	<i>Jenam Tech, LLC v. LG Electronics, Inc. et al.</i> , Case No. 4:19-cv-00249, ECF No. 1 (E.D. Tex. Apr. 3, 2019)
EX1019	Declaration of Alexa Morris for Eggert
EX1020	U.S. Pre-Grant Publication No. 2004/0093376 to De Boor <i>et al.</i>
EX1021	U.S. Patent 7,535,913 to Minami <i>et al.</i>
EX1022	U.S. Pre-Grant Publication No. 2005/0204013 to Raghunath <i>et al.</i>
EX1023	U.S. Pre-Grant Publication No. 2007/0005804 to Rideout
EX1024	U.S. Pre-Grant Publication No. 2004/0098748 to Bo <i>et al.</i>
EX1025	IETF RFC 2616 "Hypertext Transfer Protocol -- HTTP/1.1"
EX1026	U.S. Patent No. 8,259,716 to Diab
EX1027	Bova et al., RELIABLE UDP PROTOCOL <draft-ietf-sigtran-reliable-udp-00.txt> 25 February 1999
EX1028	U.S. Patent 6,674,713 to Berg et al.
EX1029	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Joint Submission of Proposed Amended Scheduling Order, ECF No. 57 (W.D. Tex. Mar. 4, 2021)
EX1030	Fourteenth Supplemental Order Regarding Court Operations Under the Exigent Circumstances Created By the COVID-19 Pandemic, The United States District Court for the Western District of Texas (March 17, 2021)
EX1031	<i>Digital Retail Apps, Inc. v. H-E-B, LP</i> , Case No. 6:19-cv-00167, Joint Stipulation and Order Postponing Trial, ECF No. 182 (W.D. Tex. Jan. 13, 2021)
EX1032	<i>Jenam Tech, LLC's Preliminary Narrowing of Claims</i> (October 20, 2020)

Petition for Post Grant Review
Patent No. 10,742,774

EX1033	<i>Jenam Tech, LLC's</i> Correspondence Regarding Narrowing of Claims (December 4, 2020)
EX1034	Judge Alan D. Albright's Case Statistics By Year (Retrieved from DocketNavigator on March 9, 2021)
EX1035	<i>Jenam Tech, LLC v. Google LLC</i> , Case No. 6:20-cv-00453, Order Granting Motion to Stay Case, ECF No. 58 (W.D. Tex. Mar. 10, 2021)
EX1036	U.S. Patent No. 10,075,564
EX1037	<i>Jenam Tech, LLC's First Set of Infringement Contentions</i> regarding U.S. Patent No. 10,742,774 (August 21, 2020)
EX1038	File History of U.S. Patent Application No. 12/714,454 (U.S. Patent No. 8,219,606)
EX1039	File History of U.S. Patent Application No. 13/477,402
EX1040	Computer-generated Redline Between U.S. Patent Application No. 12/714,454 and U.S. Patent Application No. 13/477,402
EX1041	File History of U.S. Patent Application No. 12/714,063
EX1042	Computer-generated Redline Between U.S. Patent Application No. 12/714,454 and U.S. Patent Application No. 12/714,063
EX1043	File History of U.S. Patent Application No. 15/694,802 (U.S. Patent No. 9,923,995)
EX1044	File History of U.S. Patent Application No. 14/667,642
EX1045	File History of U.S. Patent Application No. 16/040,522 (U.S. Patent No. 10,375,215)
EX1046	IETF RFC 3168 "The Addition of Explicit Congestion Notification (ECN) to IP"
EX1047	IETF RFC 6093 "On the Implementation of the TCP Urgent Mechanism"
EX1048	IETF RFC 6528 "Defending against Sequence Number Attacks"
EX1049	Excerpt of IETF webpage hosting RFC 793, retrieved May 2, 2021
EX1050	U.S. Patent No. 6,584,546 to Kavipurapu

Petition for Post Grant Review
Patent No. 10,742,774

I. INTRODUCTION

Google LLC (“Petitioner”) requests post grant review (“PGR”) of claim 1 (“the challenged claim”) of U.S. Patent No. 10,742,774 (“the ’774 patent”) (EX1001) assigned to Jenam Tech, LLC (“Patent Owner” or “PO”). For the reasons below, the challenged claim should be found unpatentable and canceled.

II. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8

A. Real Parties-in-Interest

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner identifies the following as the real parties-in-interest: Google LLC.¹

B. Related Matters

1. Related Lawsuit

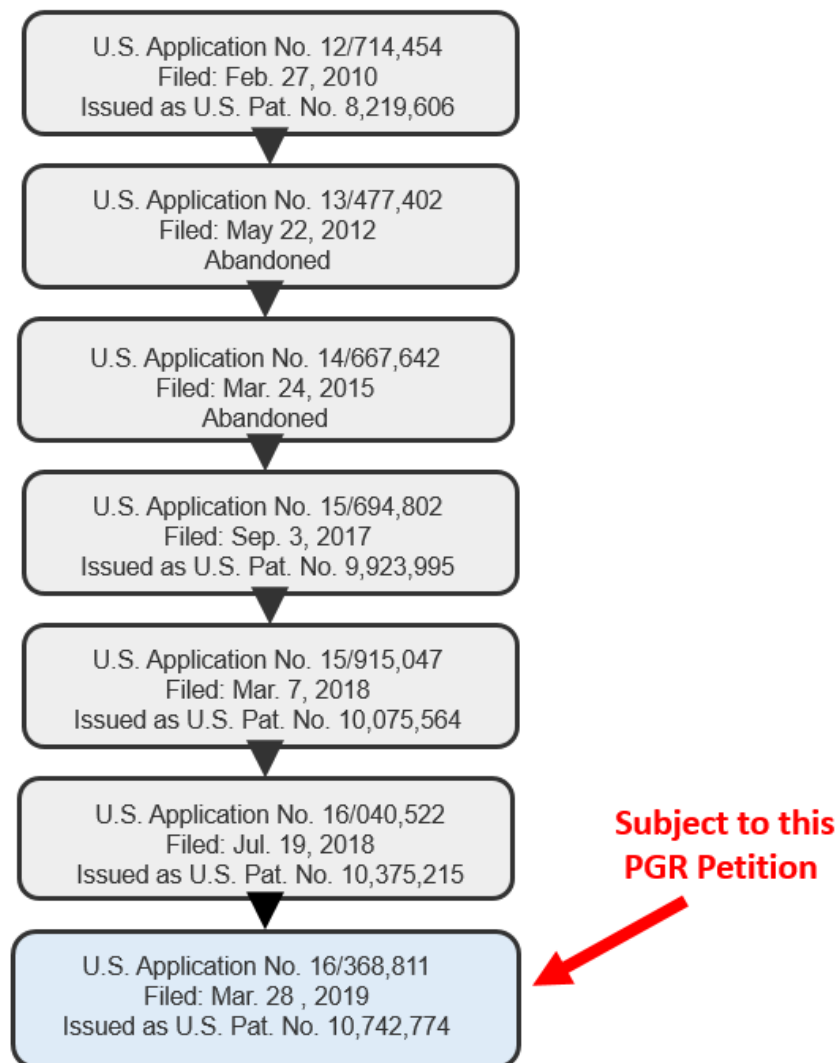
The ’774 patent is asserted in the following civil action: *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.).

2. Related Applications

The ’774 patent relates to several patents and/or patent applications, as shown in the purported priority chain below:

¹Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.

Petition for Post Grant Review
Patent No. 10,742,774



3. Related IPRs

Related U.S. Patent No. 9,923,995 (“the ’995 patent”) is subject to the following instituted IPRs: *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845 (“Google -845 IPR”); and *Unified Patents, LLC. v. Jenam Tech, LLC*, IPR2020-00742 (“Unified -742 IPR”).

Petition for Post Grant Review
Patent No. 10,742,774

Petitioner has also filed the following IPR petitions challenging patents related to the '995 and '774 patents, and which are also at issue in *Jenam Tech, LLC v. Google, LLC*, Case No. 6:20-cv-00453 (W.D. Tex.):

- IPR2021-00627 (U.S. Patent No. 10,375,215, “the ’215 patent”);
- IPR2021-00628 (U.S. Patent No. 10,075,564, “the ’564 patent”);
- IPR2021-00629 (the ’564 patent);
- IPR2021-00630 (U.S. Patent No. 10,075,565, “the ’565 patent”);
- IPR2021-00867 (U.S. Patent No. 10,306,026, “the ’026 patent”);
- IPR2021-00868 (the ’026 patent);
- IPR2021-00869 (U.S. Patent No. 10,069,945, “the ’945 patent”); and
- IPR2021-00870 (the ’945 patent).

C. Counsel and Service Information

Lead counsel: Naveen Modi (Reg. No. 46,224). Backup counsel: (1) Joseph E. Palys (Reg. No. 46,508), (2) Quadeer Ahmed (Reg. No. 60,835), (3) Jason Heidemann (Reg. No. 77,880).

Service information is Paul Hastings LLP, 2050 M Street NW, Washington, DC 20036, Tel.: 202.551.1700, Fax: 202.551.1705, email: PH-Google-Jenam-PGR@paulhastings.com. Petitioner consents to electronic service.

III. PAYMENT OF FEES UNDER 37 C.F.R. § 42.15(a)

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 50-2613.

IV. TIME FOR FILING UNDER 37 C.F.R. § 42.202

The '774 patent issued on August 11, 2020. This Petition is being timely filed no later than the date that is nine months after the date of the grant of the '774 patent.

V. GROUNDS FOR STANDING UNDER 37 C.F.R. § 42.204(a)

Petitioner certifies that the '774 patent is available for PGR and Petitioner is not barred or estopped from requesting PGR on the grounds identified herein.

As discussed below in Section IX, the '774 patent is eligible for PGR because it has at least one claim that is not entitled to a pre-AIA filing date.

VI. PRECISE RELIEF REQUESTED AND GROUNDS RAISED

A. Claim for Which Review Is Requested

Petitioner respectfully requests review of claim 1 (“challenged claim”) of the '774 patent, and cancellation of this claim as unpatentable.

B. Statutory Grounds of Challenge

The challenged claim should be canceled as unpatentable on the following grounds:

Petition for Post Grant Review
Patent No. 10,742,774

Ground 1: Claim 1 is unpatentable under AIA 35 U.S.C. § 112(b) as failing to particularly point out and distinctly claim the subject matter which the named inventor regards as the invention; and

Ground 2: Claim 1 is unpatentable under AIA 35 U.S.C. § 103 over U.S. Publication No. 2007/0171921 to Wookey *et al.* (“Wookey”) (EX1005) in view of Eggert, TCP Abort Timeout Option (April 14, 2004) (“Eggert”) (EX1006).²

The ’774 patent issued from an application filed March 28, 2019, which claims priority through a number of applications back to an application filed February 27, 2010. (EX1001, 1:8–39.) Petitioner assumes for this proceeding only, without conceding, that the earliest effective filing date of claim 1 of the ’774 patent is February 27, 2010. (*See also infra* Section IX (explaining that all claims of the ’774 patent are subject to the first-to-file provisions of the AIA).)

Wookey was published on July 26, 2007, from an application filed on November 14, 2006. (EX1005, Cover.) Therefore, *Wookey* qualifies as prior art at least under AIA 35 U.S.C. § 102(a)(1).

² The Board routinely reviews simultaneous challenges to a claim for indefiniteness and based on prior art. (*See infra* Section XI.A.2.b.)

Petition for Post Grant Review
Patent No. 10,742,774

Eggert, which is an Internet Engineering Task Force (IETF) Internet-Draft (or “ID”) working document, was published on April 14, 2004. (EX1006, 1; *see also infra* Section XII.) As confirmed by the declaration of Alexa Morris, Managing Director of IETF, *Eggert* was published, disseminated, and reasonably available to the public by April 15, 2004. (EX1019, ¶¶9–10; *see also infra* Section XII.) Thus, *Eggert* qualifies as prior art at least under AIA 35 U.S.C. § 102(a)(1).

Wookey was not considered during prosecution of the ’774 patent (EX1001, Cover (“References Cited”); *see also generally* EX1004) and is not the same or substantially similar to any art previously presented to the Office. While *Eggert* was cited in an IDS during prosecution, 35 U.S.C. § 325(d) does not provide a basis for discretionary denial, as discussed below. (*See infra* Section XIII.A.)

VII. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art at the time of the alleged invention of the ’774 patent (“POSITA”) would have had an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking. (EX1002 ¶¶16–18.)³ More education can supplement practical experience and vice versa. (*Id.*)

³ Petitioner submits the declaration of Dr. Bill Lin (EX1002), an expert in the field of the ’774 patent. (EX1002 ¶¶3–15; EX1003.)

VIII. OVERVIEW OF THE '774 PATENT

The '774 patent is directed to networking, and, to the sharing of information for detecting an idle TCP connection. (EX1001, 2:37–39, 8:44–65.) Figure 7 illustrates such a process.

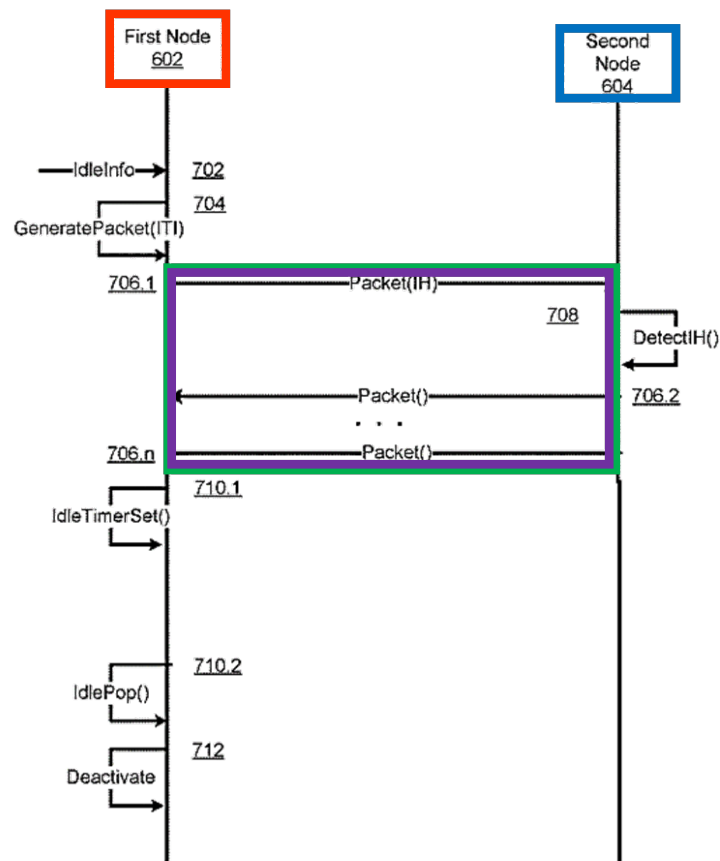


Fig. 7

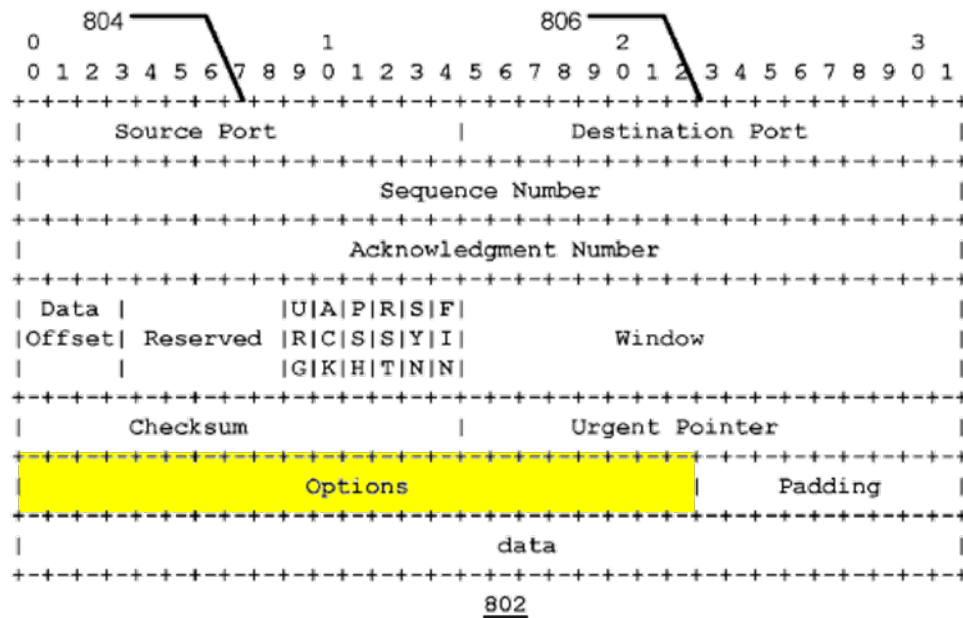
(*Id.*, FIG. 7 (annotated); EX1002 ¶64.)

First, the **first node 602** receives a message 702 identifying idle information representing a duration for an idle time period (“ITP”). (EX1001, 12:3–5.) The idle

Petition for Post Grant Review
Patent No. 10,742,774

information “may include and/or identify a duration of time for detecting an idle time period.” (*Id.*, 12:51–55.) The “duration may be specified according to various measures of time[,] including seconds.” (*Id.*; EX1002 ¶65.)

Next, the **first node** “generat[es] a TCP packet including an ITP header based on received idle information.” (*Id.*, 16:17–20.) The **TCP options field** of a TCP packet may store this ITP header. (*Id.*, 15:38–58.) Figure 8 below, “adapted from RFC 793,” illustrates this configuration. (*Id.*, FIG. 8, 6:42–44, 15:38–58.)



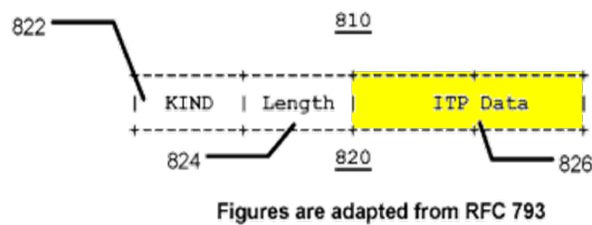


Fig. 8

(*Id.*, FIG. 8 (cropped and annotated); EX1002 ¶67.)

The ITP header is exchanged during the three-way handshake. (EX1001, 14:63–15:10.) For example, as shown in Figure 7, the **first node** transmits a message 706.1, i.e., a TCP packet including an ITP header (IH) that contains ITP information, to **second node 604**. (*Id.*, 16:42–44.) Message 708 exemplifies the **second node**’s detection of the IH in the received TCP packet. (*Id.*, 21:62–67; EX1002 ¶68.)

The ’774 patent explains that “a TCP keep-alive option, a TCP user timeout, a retransmission timeout, an acknowledgment timeout, and/or another timeout associated with a TCP connection may be modified based on the first idle information.” (EX1001, 13:62–65; EX1002 ¶69.)

Claim 1 recites limitations relating to the above features but in context of “TCP-variant” packet and connection. (EX1001, claim 1.) However, the ’774 patent describes its features in terms of a TCP connection and packet and never uses the term “TCP-variant” or explains the use of a “TCP-variant” packet or connection in the way claimed in claim 1. (*See infra* Section XI.A; *see also generally* EX1001,

Petition for Post Grant Review
Patent No. 10,742,774

2:43–6:6 (Summary discussing aspects relating to a “TCP” packet and “TCP” connection)⁴, 6:48–24:20 (Detailed Description section discussing aspects in terms of a “TCP” packet and “TCP” connection)⁵; EX1002 ¶70.)

Moreover, as explained below, all the limitations in claim 1 were known in the prior art and obvious. (*See* Section XI.B; EX1002 ¶71; *see also id.* ¶¶19–63 (discussing technology background), citing, e.g., EXS. 1007–1009, 1011–1015, 1020–1028.)

IX. THE '774 PATENT IS ELIGIBLE FOR PGR

The PGR provisions of the Leahy-Smith America Invents Act, Pub. L. No. 112–29, 125 Stat. 284 (2011) (“AIA”) apply to patents subject to the first inventor to file provisions of the AIA, i.e., patents having at least one claim with an effective

⁴ While the Summary section also refers to a second protocol that is “separate” or “different” from “the TCP” (EX1001, Abstract, 3:6–7, 3:39–40, 4:9–10, 4:50–51), the detailed description provides no disclosures of these features at all. (*Id.*, 6:48–24:20; EX1002 ¶70 n.2.)

⁵ As discussed further below, the specification does mention a “variant of the current TCP,” but does so in terms of the location and structure of an ITP header that “may be included in a TCP packet.” (EX1001, 16:1–7; *see infra* Section XI.A; EX1002 ¶70 n.3.)

Petition for Post Grant Review
Patent No. 10,742,774

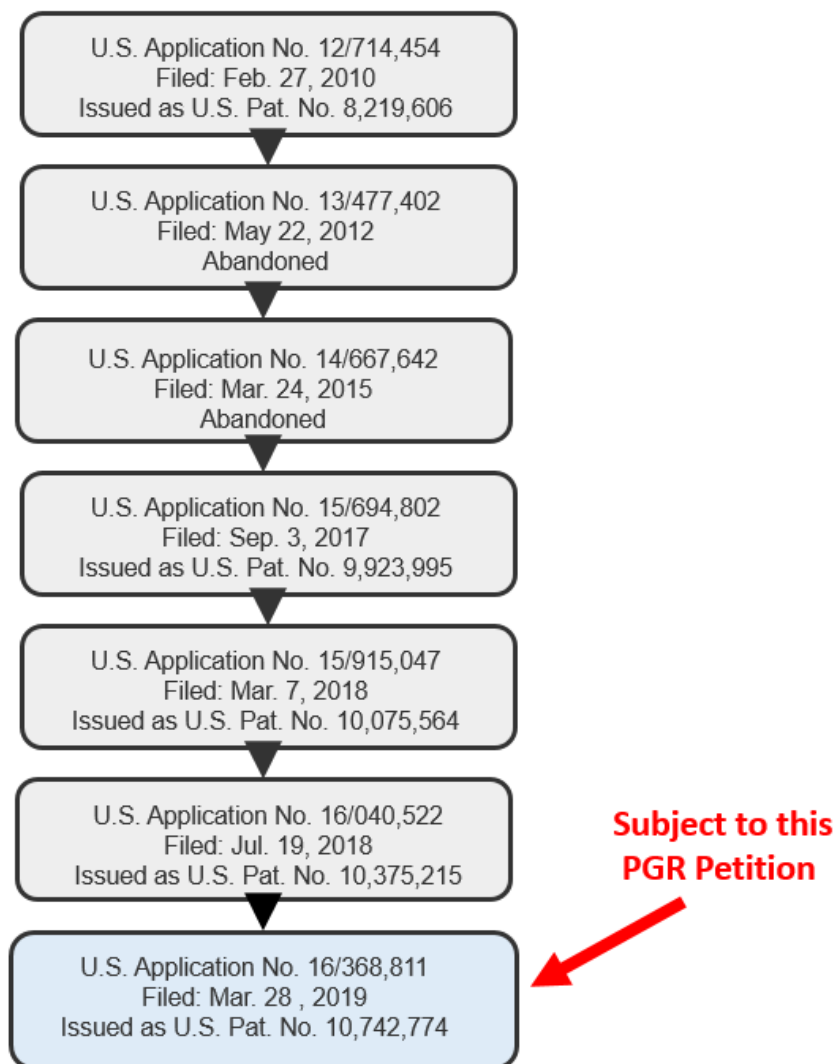
filing date on or after March 16, 2013. *Grunenthal GmbH v. Antecip Bioventures II LLC*, PGR2018-00001, Paper 17 at 9–10 (May 1, 2018). A claim in a U.S. patent application is entitled to the benefit of the filing date of an earlier filed U.S. or PCT application if the subject matter of the claim is disclosed in the earlier filed application in accordance with the written description requirement. *PowerOasis, Inc. v. T-Mobile USA, Inc.*, 522 F.3d 1299, 1306 (Fed. Cir. 2008) (subject matter disclosed for first time in a continuation application does not receive benefit of the parent’s filing date); *see also In re Gosteli*, 872 F.2d 1008, 1010–11 (Fed. Cir. 1989).

To comply with the written description requirement, the specification or earlier-filed application “must describe the invention sufficiently to convey to a person of skill in the art that the patentee had possession of the claimed invention at the time of the application, i.e., that the patentee invented what is claimed.” *LizardTech, Inc. v. Earth Resource Mapping, Inc.*, 424 F.3d 1336, 1345 (Fed Cir. 2005); *see also Lockwood v. Am. Airlines, Inc.*, 107 F.3d 1565, 1572 (Fed. Cir. 1997); *Allergan, Inc. v. Sandoz Inc.*, 796 F.3d 1293, 1308–09 (Fed. Cir. 2015). “The test requires an objective inquiry into the four corners of the specification from the perspective” of a POSITA. *Ariad Pharms., Inc. v. Eli Lilly and Co.*, 598 F.3d 1336, 1351 (Fed. Cir. 2010) (en banc). Whether the added subject matter is an obvious variant of the disclosed subject matter is irrelevant. *Lockwood*, 107 F.3d at 1572 (“It is not sufficient for purposes of the written description requirement of § 112 that the

Petition for Post Grant Review
Patent No. 10,742,774

disclosure, when combined with the knowledge in the art, would lead one to speculate as to modifications that the inventor might have envisioned, but failed to disclose.”).

The '774 patent relates to several patents and/or patent applications, as shown in the purported priority chain below:



(See also *supra* Section II.B.2.)

Petition for Post Grant Review
Patent No. 10,742,774

As the above chart shows, the '774 patent relates to two applications filed before March 16, 2013, namely U.S. Patent Application No. 12/714,454 (“the '454 application”) (EX1038) and U.S. Patent Application No. 13/477,402 (“the '402 application”) (EX1039).

The '774 patent is eligible for PGR because it has at least one claim that is not entitled to the filing date of either the '454 application or the '402 application (“the pre-AIA applications”). *See Inguran, LLC d/b/a/ Sexing Technologies v. Premium Genetics (UK) Ltd.*, PGR2015-00017, Paper 8 (Institution Decision) at 9–11 (December 22, 2015).

For example, as explained below, at least claim 42 of the '774 patent includes subject matter that is not disclosed in the pre-AIA applications. Therefore, claim 42 is not entitled to the filing date of either of the pre-AIA applications and thus is not entitled to a pre-March 16, 2013 filing date. *PowerOasis, Inc.*, 522 F.3d at 1306; *In re Gosteli*, 872 F.2d at 1010–11. (EX1002 ¶¶94–119.) Accordingly, the '774 patent is eligible for PGR.

A. The Pre-AIA Applications Do Not Support the “Non-TCP” Features Recited in Claim 42

Independent claim 42 of the '774 patent recites features concerning a “non-TCP connection” and a “non-TCP packet.” Specifically, claim 42 recites a method comprising the following limitations:

Petition for Post Grant Review
Patent No. 10,742,774

- “receiving information for use in: detecting a time period that results in a ***non-TCP connection*** being subject to at least partial deactivation...” (EX1001, 29:12–14);⁶
- “sending a ***non-TCP packet*** that is based on the information and that includes a time period parameter field identifying metadata” (*id.*, 29:14–17); and
- “sending, to a second node and for setting up the ***non-TCP connection***, the ***non-TCP packet*** to provide the metadata to the second node, for use by the second node in determining a timeout attribute associated with the ***non-TCP connection***” (*id.*, 29:18–23).

Claim 67, which depends from claim 43, which in turn depends from claim 42, further explains that “***the non-TCP connection***” of claim 42 “***is not*** a TCP-extension.” (*id.*, 33:47–48 (claim 67), 29:24–30 (claim 43).)

As demonstrated in turn below, the two pre-AIA applications do not describe a “non-TCP connection” or “non-TCP packet” in the manner recited by claim 42, and the material incorporated by reference in the two pre-AIA applications also does not cure this deficiency. (*See generally* EX1038 (‘454 application), 99–106 (abstract

⁶ Emphasis added unless otherwise noted.

and claims) 111–118 (figures), 120–165 (specification); EX1039 ('402 application), 85 (abstract), 90–103 (claims and figures), 106–151 (specification).)

1. The Pre-AIA '454 Application Does Not Support the “Non-TCP” Features

The disclosure of the '454 application (including the claims) only describes TCP connections (including extensions and variants thereof) but does not describe to a POSITA a “non-TCP connection” as required by claim 42 of the '774 patent. (EX1002 ¶¶97–108.)

In particular, the '454 application uses the term “TCP” nearly 400 times including, for instance, in the specification (310 times), claims (72 times), abstract (3 times), and figures (14 times). (*See generally* EX1038 ('454 application), 99–106, 111–118, 120–165.) Not once does the '454 application refer to a “non-TCP” connection or packet (or variants thereof). (*Id.*; EX1002 ¶98.)

For example, both the Abstract and “Summary” of the as-filed '454 application describes the disclosed invention as relating to a “***TCP connection***” and “***TCP packet***” (as opposed to a ***non-TCP*** connection or packet required by claim 42):

Methods and systems are described for sharing information for detecting an idle *TCP connection*. In one aspect, a method includes receiving, by a second node from a first node, a first *transmission control protocol (TCP) packet* in a *TCP connection*. The method further includes ... identifying metadata for a first idle time period, detectable by the first node, during which no *TCP packet* including data in a first TCP data stream sent in the *TCP connection* by the second node is received by the first node. The method still further includes modifying ... a timeout attribute associated with the *TCP connection*.

(EX1038, 122 (¶[0008]), 99 (Abstract) (describing same); *see also id.*, 122–124 (¶¶[0009]–[0011] (describing additional aspects in the Summary in context of a “TCP packet” and “TCP connection”))).)

The detailed description in the ’454 application likewise focuses on a “TCP connection.” (*Id.*, 125–126 (¶¶[0013]–[0020].) For example, with reference to figures 2–4b, the ’454 application explains:

FIG. 2 is a flow diagram illustrating a first method for sharing information for detecting an idle **TCP connection** according to an exemplary aspect of the subject matter described herein. FIG. 3 is a flow diagram illustrating a second method for sharing information for detecting an idle **TCP connection** according to an exemplary aspect of the subject matter described herein. FIG. 4a is a block diagram illustrating a system for sharing information for detecting an idle **TCP connection** according to the first method in FIG. 2. FIG. 4b is a block diagram illustrating a system for sharing information for detecting an idle **TCP connection** according to the second method in FIG. 3.

(*Id.*, 131–132 (¶[0035]); *see also e.g., id.* (133–137 (¶[0042]–[0051] (describing in connection with figure 5, “TCP packets” and “TCP connection” and “TCP layer” features), 141 (¶[0064] (discussing a “TCP connection” as one that “may be identified by its endpoints” and “may include an endpoint of the TCP connection”), 143 (¶[0069] (discussing a “TCP keep-alive option, a TCP user timeout” and “another timeout associated with a TCP connection”); EX1002 ¶[99–100.]

While the ’454 application discusses that “[a]n equivalent or analog of an ITP header may be included in a footer of a protocol packet in **an extension and/or variant of the current TCP**” (EX1038, 147 (¶[0080])), such extensions or variants are mentioned solely within the context of a TCP packet:

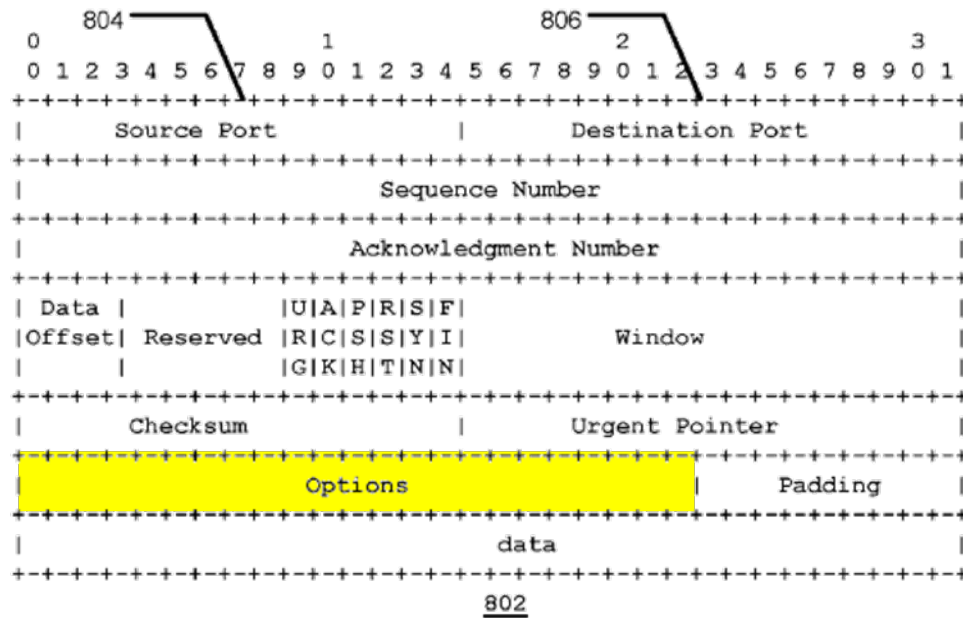
Those skilled in the art will recognize given this disclosure that an *ITP header* may have other suitable formats and *may be included in a TCP packet in structures and locations other than those specified for TCP options in RFC 793*. An equivalent or analog of an ITP header may be included in a footer of a *protocol packet in an extension and/or variant of the current TCP*.

(*Id.*, 147 (¶[0080]); EX1002 ¶101.)

Indeed, to the extent any embodiment is disclosed in relation to these features, it relates to using a TCP options field in a conventional TCP packet structure provided by RFC 793 to share idle time information between nodes. (EX1038, 15:38–58.)⁷ Figure 8 below, “adapted from RFC 793,” illustrates this configuration. (*Id.*, FIG. 8, 6:42–44, 15:38–58.)

⁷ RFC 793 is titled “Transmission Control Protocol, DARPA Internet Program Internet Protocol Specification.” (EX1008, 1; EX1001, 1:61–68.)

Petition for Post Grant Review
Patent No. 10,742,774



(*Id.*, FIG. 8 (cropped and annotated); EX1002 ¶102; *see also* EX1008, FIG. 3.)

Thus, the disclosed extension or variant of “the current TCP” in the ’454 application does not describe any non-TCP aspects, especially when viewed in context of surrounding language in the specification that, as explained above, is entirely focused on TCP-based features to the absence of any “non-TCP” features. (EX1002 ¶103.)

The understanding that “an extension ... of the current TCP” disclosed in ¶[0080] above cannot encompass a “non-TCP” feature is consistent with claim 67

Petition for Post Grant Review
Patent No. 10,742,774

of the '774 patent, which makes clear that “*the non-TCP connection*” of claim 42 “*is not* a TCP-extension.” (*Id.*, 33:47–48 (claim 67).)⁸

Likewise, the '774 patent confirms that the “variant of the current TCP” disclosed in ¶[0080] above also cannot encompass the claimed “non-TCP” connection or packet features because it includes claims that separately recite “TCP-variant” and “non-TCP.” (*E.g.*, *Compare* EX1001, 24:22–41 (TCP-variant based claim 1) *with* 29:10–23 (non-TCP based claim 42); EX1002 ¶¶106–107.) The '774 patent’s separate use of these different claim terms demonstrates the distinction the patent makes between a “variant” of TCP and “non-TCP.” Indeed, Patent Owner similarly differentiated between the “TCP-variant” and “non-TCP” terms in other

⁸ To the extent it is argued that claim 67 demonstrates that the “non-TCP connection” in claim 42 should be broad enough to encompass a TCP-extension pursuant to the doctrine of claim differentiation, that interpretation would be unreasonable and unsupported by the specification for the reasons explained. Moreover, the scope of “non-TCP connection” in the '774 patent under such an interpretation would then have to include things that are beyond a TCP extension, which is not described anywhere in the specifications, drawings, and figures of the pre-AIA applications. (EX1002 ¶105).

Petition for Post Grant Review
Patent No. 10,742,774

patents from the '774 patent family. (*See e.g.*, EX1010 ('995 patent), 23:4–21 (TCP-variant-based claim 1), 24:33–51 (TCP-variant-based claim 15), 26:7–14 (non-TCP-based claim 27), 26:43–28:16 (non-TCP-based claims 29–30); EX1036 ('564 patent), 23:5–21 (TCP-variant-based claim 1), 24:29–47 (TCP-variant-based claim 16), 25:27–53 (non-TCP-based claim 24), 26:6–29 (non-TCP-based claim 28).)

Thus, for the reasons above, the '454 application does not provide any support for the “non-TCP connection” and “non-TCP packet” features recited in claim 42 of the '774 patent. (EX1002 ¶108.)

2. The Pre-AIA '402 Application Does Not Support the “Non-TCP” Features

The '402 application is a continuation of the '454 application (EX1001, 1:30–39; EX1039 ('402 application), 87, 106 (¶[0001])). Therefore, substantively, the '402 application includes the same disclosure as the '454 application. (EX1040 (computer-generated redline showing the similarities in BLACK and differences between the '454 and '402 applications (i.e., BLUE for what is only present in the '402 application and RED for what is only present in the '454 application))).) *See e.g., Applied Materials, Inc. v. Advanced Semiconductor Materials Am., Inc.*, 98 F.3d 1563, 1579–80 (Fed. Cir. 1996), *cert. denied*, 520 U.S. 1230 (1997) (“Although there may be some variation in the scope of the claimed subject matter, a continuation application is based solely on the disclosure of a parent application ...

Petition for Post Grant Review
Patent No. 10,742,774

By definition, a continuation adds no new matter and is akin to an amendment of a pending application.”) (Mayer, J., concurring). To be sure, just like the ’454 application, the ’402 application’s specification, figures, abstract, and as-filed claims do not describe any “non-TCP connection” or “non-TCP packet” features. (*See generally* EX1039 (’402 application); EX1002 ¶¶109–111.)

Thus, for the same reasons explained above for the ’454 application, the ’402 application does not provide any support for the “non-TCP connection” and “non-TCP packet” features recited in claim 42 of the ’774 patent. (EX1002 ¶112.)

3. Materials Incorporated by Reference in the Two Pre-AIA Applications Do Not Support the “Non-TCP” Features

The two pre-AIA applications purport to incorporate U.S. Application No. 12/714,063 (“the ’063 application”) (EX1041) and RFC 793 (EX1008). (EX1038 (’454 application), 120 (¶¶[0001]–[0002]), 90 (IDS citing RFC 793); EX1039 (’402 application), 106 (¶¶[0002]–[0003]); EX1002 ¶¶113–119.) However, as with the pre-AIA applications, neither the ’063 application nor RFC 793 disclose or mention the “non-TCP connection” or “non-TCP packet” features recited in claim 42 of the ’774 patent.

The ’063 application shares a similar specification to the ’454 application and similarly only describes “TCP connections” and “TCP packets” (including extensions and variants in similar fashion as the ’454 application). (*See generally*

Petition for Post Grant Review
Patent No. 10,742,774

EX1041, 137–201; *see also* EX1042 (computer-generated redline showing the similarities in BLACK and differences between the as-filed specifications in the '063 and '454 applications (i.e., BLUE for what is only present in the '063 application and RED for what is only present in the '454 application).) (EX1002 ¶115.)

RFC 793 dates back to 1981 (EX1008, 1) and defined TCP at that time (*id.*). Indeed, RFC 793 provides:

This document describes the DoD *Standard Transmission Control Protocol (TCP)*. There have been nine earlier editions of the ARPA TCP specification on which this standard is based, and the present text draws heavily from them.

(EX1008, 4.) However, RFC 793 likewise does not describe the “non-TCP connection” and “non-TCP packet” features like that recited in claim 42. (EX1002 ¶116.)

Thus, even if the material disclosed in RFC 793 and/or the '063 application are incorporated into the pre-AIA application disclosures, none of the material incorporated by reference supports the claimed “non-TCP connection” and “non-TCP packet” features recited in claim 42 of the '774 patent. (EX1002 ¶117.)

Therefore, neither of the two pre-AIA applications convey to a POSITA that the named inventor had possession of the “non-TCP connection” and/or “non-TCP packet” features recited in claim 42 at the relevant time. As explained, neither pre-

Petition for Post Grant Review
Patent No. 10,742,774

AIA application (or its purported incorporated material) describes a “non-TCP connection” and/or a “non-TCP packet” in any respect, let alone in the context of the claimed “detecting a time period that results in a *non-TCP connection* being subject to at least partial deactivation,” “sending a *non-TCP packet* that is based on the information and that includes a time period parameter field identifying metadata,” or “sending, to a second node and for setting up *the non-TCP connection, the non-TCP packet* to provide the metadata to the second node, for use by the second node in determining timeout attribute associated with *the non-TCP connection*,” as recited in claim 42 of the ’774 patent. (EX1002 ¶118.)

The above understanding is consistent with the prosecution history of the ’774 patent family, as the terms “non-TCP connection” and/or a “non-TCP packet” appear in the family of applications for the ’774 patent for the first time in U.S. Patent Application No. 15/694,802 (“the ’802 application”) (EX1043) filed September 3, 2017. The ’802 application is a continuation-in-part application of U.S. Patent Application No. 14/667,642 (“the ’642 application”) (EX1044), which is a

Petition for Post Grant Review
Patent No. 10,742,774

continuation-in-part application of the '402 application.⁹ The '802 application introduced the non-TCP connection and non-TCP packet terms only in its Summary section of the specification and in the claims.¹⁰ (EX1043, 17–18 (¶¶[0012]–[0013],

⁹ The '642 application only mentions “TCP” (EX1044, 126 (¶[0048]), *see also id.*, 109 (¶¶[0003]–[0004], 162–164 (¶[0128]), and does not refer to a “non-TCP” connection or packet. (EX1002 ¶119 n.6.)

¹⁰ Some of the claims reciting “non-TCP connection” in the '802 application are now found in the '995 patent, which issued from the '802 application, and are challenged in the Unified -742 IPR. (*See supra* Section II.B.3.) In the Unified -742 IPR, the Board noted that “[a]lthough the '995 patent claims priority to applications filed before March 16, 2013, Patent Owner has not shown that the written description of the earlier applications supports the challenged claims.” Unified -742 IPR, Paper 11 (Institution Decision) at 4 n.1 (October 8, 2020). Patent Owner responded that “without waiving any right to claim an earlier priority date, and contest the applicable patent laws governing the '995 patent, Patent Owner hereby addresses the Petition and Board’s Institution Decision under post-AIA patent law.” Unified -742 IPR, Paper 13 (Patent Owner Response) at 2 n.2 (January 1, 2021).

Petition for Post Grant Review
Patent No. 10,742,774

70–77; *see also id.*, 100 (substitute specification contains no new matter), 105–106 (¶¶[0012]–[0013]); EX1002 ¶119.)

For the reasons above, claim 42 of the ’774 patent is not entitled to an effective filing date earlier than the September 3, 2017 filing date of the ’802 application. (EX1002 ¶¶94–119.)

B. AIA Applicability

The ’774 patent is eligible for PGR. As discussed above, at least claim 42 of the ’774 patent includes subject matter that is not disclosed by an application filed before March 16, 2013. Therefore, every claim of the ’774 patent is subject to the first-to-file provisions of 35 U.S.C. § 102(a). *See Inguran, LLC d/b/a/ Sexing Technologies*, PGR2015-00017, Paper 8 (Institution Decision) at 9–11.

X. CLAIM CONSTRUCTION

In a post grant review, claims are construed in accordance with the ordinary and customary meaning of such claims as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent. 37 C.F.R. § 42.200(b). In particular, claim terms are generally given their “ordinary and customary meaning,” that is, “the meaning that the term would have to a POSITA in question at the time of the invention, i.e., as the effective filing date of the patent application.” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005) (*en banc*). In the case that “the specification . . . reveal[s] a special definition given to a claim term by the patentee

Petition for Post Grant Review
Patent No. 10,742,774

that differs from the meaning it would otherwise possess . . . the inventor’s lexicography governs.” *Id.* at 1316 (internal citation omitted).

The Board only construes claims when necessary to resolve the underlying controversy. *Toyota Motor Corp. v. Cellport Systems, Inc.*, IPR2015-00633, Paper 11 at 16 (August 14, 2015). Petitioner submits that, for purposes of this proceeding, no term requires construction.¹¹ (EX1002 ¶73.)

Challenged claim 1 recites the terms “TCP-variant packet” and “TCP-variant connection” (*see, e.g.*, EX1001, 24:27–30; *see also id.*, 24:31–41), which are also recited in claims of the related ’995 patent at-issue in the Unified -742 IPR and Google -845 IPR (*see supra* Section II). The ’774 patent and the ’995 patent share a common specification and drawings. (*Compare* EX1001 *with* EX1010 (’995 patent).)

Like the Board noted for the ’995 patent, the specification “provides little guidance as to [the] meaning [of ‘TCP-variant connection’].” Google -845 IPR,

¹¹ Petitioner reserves all rights to raise claim construction and other arguments in district court as relevant to those proceedings. A comparison of the claims to any accused products in litigation may raise controversies that are not presented here given the similarities between the references and the patent.

Petition for Post Grant Review
Patent No. 10,742,774

Paper 16 at 17 (October 8, 2020); Unified -742 IPR, Paper 11 at 5 (October 8, 2020). (*See generally* EX1001.) For these and other reasons discussed below, the “TCP-variant” terms render claim 1 indefinite. (*See infra* Section XI.A.) To the extent the scope of the claimed “TCP-variant” features in claim 1 can be somewhat understood under their plain meaning (which is insufficient to make the claim definite), such features are disclosed and/or suggested by *Eggert* like the Board preliminarily found in the Google -845 IPR and Unified -742 IPR. (*See infra* Section XI.B.)

XI. DETAILED EXPLANATION OF GROUNDS**A. Ground 1: Claim 1 Is Indefinite**

“[A] patent’s claims, viewed in light of the specification and prosecution history, [must] inform those skilled in the art about the scope of the invention with reasonable certainty.” *Nautilus, Inc. v. Biosig Instr., Inc.*, 572 U.S. 898, 901 (2014).¹² The requirement is aimed at providing the public with clear notice about

¹² *See also* Memorandum, *Approach to Indefiniteness Under 35 U.S.C. § 112 in AIA Post-Grant Proceedings* (Jan. 6, 2021, available at

<https://www.uspto.gov/sites/default/files/documents/IndefinitenessMemo.pdf>)

(Board applies the *Nautilus* standard when determining indefiniteness in AIA post-grant proceedings). Even if the Board were to apply the pre-*Nautilus* standard (the

Petition for Post Grant Review
Patent No. 10,742,774

what is being claimed. *Interval Licensing LLC v. AOL, Inc.*, 766 F.3d 1364, 1370 (Fed. Cir. 2014) (quoting *Nautilus*, 572 U.S. at 909).

Claim 1 of the '774 patent fails to provide such notice as it does not inform a POSITA about the scope of the claimed "TCP-variant" features. Namely, the claim language, specification, and file history do not provide any guidance as to the bounds of the claimed "TCP-variant connection" and "TCP-variant packet" terms with reasonable certainty. (*See infra* Section XI.A.1; EX1002 ¶¶120–122.) For example, there are some packets/connections that are within the scope of the "TCP-variant" terms such as the use of an options field in a TCP packet defined by RFC 793, as disclosed by the '774 patent and *Eggert*. (*See supra* Sections VIII, X, *infra* Sections XI.A.2.b, XI.B.) However, the lack of clarity in the '774 patent regarding the "TCP-variant" terms raises unanswered questions regarding the bounds of these

In re Packard standard), where a claim is held to be indefinite when it contains words or phrases whose meaning is unclear in describing the claimed invention, the result would have been the same because, as discussed below, the claims include words or phrases whose meaning is unclear. *See Nippon Suisan Kaisha Ltd. v. Pronova Biopharma Norge AS*, PGR2017-00033, Paper 37 at 11-12, 14 (January 16, 2019), citing *In re Packard* 751 F.3d 1307, 1311 (Fed. Cir. 2014).

Petition for Post Grant Review
Patent No. 10,742,774

terms, such as what baseline “TCP” to use for measuring the variance (*see infra* Section XI.A.2.a) and what qualifies as a “variant” from TCP including what type of variance and how much (or little) variance from the undefined baseline “TCP” is required to constitute a “TCP-variant” (*see infra* Section XI.A.2.b). Other claims of the ’774 patent and claims in the ’774 patent family further demonstrate the ambiguity introduced by the “TCP-variant” terms. (*See infra* Section XI.A.2.c.) Thus, as explained below, because claim 1 fails to particularly point out and distinctly claim the subject matter regarded as the invention, it is indefinite under 35 U.S.C. § 112(b).

1. The Claim Language, Specification, and File History Do Not Sufficiently Describe the “TCP-Variant” Features

The “TCP-variant” term found in claim 1 is recited in the context of a “TCP-variant” packet and connection:

1. An apparatus comprising:

... wherein the one or more processors execute the instructions for:

receiving, by a second node from a first node, a *transmission control protocol (TCP)-variant packet* in advance of a *TCP-variant connection* being established;

detecting a time period parameter field in the *TCP-variant packet*;

identifying metadata in the time period parameter field for a time period, during which no packet is received from the first node in a data stream of the *TCP-variant connection* and processed by the second node to keep the *TCP-variant connection* active; and

calculating, by the second node and based on the metadata, a timeout attribute associated with the *TCP-variant connection* for being used to at least partially close the *TCP-variant connection*.

(EX1001, 24:22–41.) Other than using the term, claim 1 and its dependents do not sufficiently describe what is meant by a *variant* of a TCP connection or packet. (*Id.*; EX1002 ¶123.) Nor does the specification provide the requisite clarity.

Petition for Post Grant Review
Patent No. 10,742,774

The '774 patent specification describes features for a TCP (not a TCP-variant). For example, the specification addresses an alleged “need...for sharing information for detecting an idle TCP connection.” (EX1001, 2:37–39.) Indeed, the patent’s “Summary” discloses methods and systems concerning an “idle TCP connection” that refer to a “TCP connection” and “TCP packet.” (*Id.*, 2:40–6:6.) The detailed description does the same. As explained above, the description describes a process for detecting an idle **TCP connection** (not a TCP-variant one). (*Supra* Section VIII; EX1001, FIG. 7, 8:66–9:13, 12:12–18, 12:40–45; EX1002 ¶124.) The patent also describes a “TCP packet” including an “ITP header” that “may be included in a packet exchanged during setup of TCP connection” (EX1001, FIG. 8, 14:63–66) that has a configuration “adapted from RFC 793” (*id.*, 15:38–67, FIG. 8). (EX1002 ¶124.)

The *only* place where the specification mentions a TCP variant concept is in the context of the ITP header of a TCP packet:

Those skilled in the art will recognize given this disclosure that ***an ITP header may have other suitable formats and may be included in a TCP packet in structures and locations other than those specified for TCP options in RFC 793.*** An equivalent or analog of an ITP header may be included in a footer of a protocol packet in an extension and/or ***variant of the current TCP.***

(EX1001, 16:1–8; EX1002 ¶125.)

Petition for Post Grant Review
Patent No. 10,742,774

The above disclosure (or any other portion of the specification) does not provide any discussion or details that would inform a POSITA with reasonable certainty as to the scope of a “TCP-variant packet” (e.g., how much variance from a TCP packet is permitted and what types of packets are included within the scope of a TCP-variant packet). (EX1002 ¶126.) Nor does any portion of the specification provide any details that would inform a POSITA with reasonable certainty as to the scope of a “TCP-variant connection.”¹³ (*Id.*)

¹³ Indeed, the whole concept of what qualifies as a “variant” in the context of the specification leaves questions as to its proper scope. For example, a POSITA would not have been informed with reasonable certainty whether a “variant” of TCP requires a modification of existing functionality from what is done in “the current TCP” (which itself is undefined as discussed below). (EX1002 ¶126 n.7.) If not, a POSITA would not have been informed with reasonable certainty whether a protocol including additional functionality not present in the undefined “current TCP” qualifies as a “variant” of the undefined “current TCP” or an “extension” of the undefined “current TCP.” (*Id.*) Such requisite guidance, which would have informed the scope of the “TCP-variant” features, is altogether missing from the ’774 patent disclosure as discussed below in Section XI.A.2.

The file history of the '774 patent (and its relevant parent applications) also does not offer any assistance in ascertaining what is meant by the claimed “TCP-variant” features. (*See generally* EX1004; EX1043; EX1045.) If anything, as discussed below, other claims pursued during prosecution (now issued) further demonstrate the indefiniteness of the claimed “TCP-variant” features. (*See infra* Section XI.A.2.c; EX1002 ¶127.)

2. The Intrinsic Record Does Not Inform a POSITA About the Scope of the Claimed “TCP-Variant” Features Recited With Reasonable Certainty

Claim 1 is indefinite for several reasons. First, the intrinsic record for the '774 patent does not explain, for the “TCP-variant” features in claim 1, what baseline “TCP” to use from which the “varian[ce]” should be measured. Second, the intrinsic record does not explain what type of (or how much or how little) “variance” from the “TCP” would encompass the claimed “TCP-variant” features. Third, other claims in the '774 patent further demonstrate the ambiguity introduced by the “TCP-variant” terms and confirm that a POSITA would not be able to ascertain with reasonable certainty the scope of the “TCP-variant” features. (EX1002 ¶¶128–152.)

a) **The Intrinsic Record Does Not Inform About the Scope of the [Current] TCP From Which the Claimed TCP-Variant Features Must Be Considered**

To understand with reasonable certainty the scope of the claimed “TCP-variant” features, a POSITA would need to understand the scope of the “TCP” from which the variance is based. (EX1002 ¶¶129–134.) Here, while the specification mentions “variant of the *current TCP*” (EX1001, 16:4–7), the ’774 patent lacks guidance in determining the scope of claim 1 with reasonable certainty because the ’774 patent does not describe what is meant by “current TCP.” (EX1002 ¶129.) Namely, the ’774 patent specification does not identify what the “current TCP” is, or explain any mechanism on how to identify what is or is not a “current TCP,” and thus in turn, what is a “variant of the current TCP.” (*Id.*)

The ’774 patent specification states there are “various implementations” of TCP, with no further discussion or support (EX1001, 1:55–58), and also references RFC 793 (*id.*, 1:61–67) to provide details on TCP functionality. (*Id.*, 11:47–48 (“Detailed information on the operation of TCP is included in RFC 793”), 14:63–66, 15:13–14 (“Such other exchanges are not currently supported by the TCP as described in RFC 793.”), 15:42, 16:1–4, 20:51–57.) As such, the specification (and claims) do not inform a POSITA with reasonable certainty as to whether only components/characteristics specified in RFC 793 define the “current TCP” described in the specification and “TCP” in claim 1 (EX1001, 16:1–8, Claim 1

Petition for Post Grant Review
Patent No. 10,742,774

(“*TCP*-variant”)) or whether the “TCP” in “TCP-variant” encompasses other unspecified implementations of TCP at the time. (EX1002 ¶130.)

RFC 793, which is dated September 1981 (EX1008, 1), predates the filing of the application for the ’774 patent by over 37 years. (EX1001, cover.)¹⁴ Moreover, in addition to the “*various* implementations” of TCP acknowledged by the ’774 patent, TCP implementations were defined by way of other RFC documents published after RFC 793 and before February 2010 (the earliest filing date to which the ’774 patent purports to claim priority), such as RFC 1122 (EX1009) and RFC 3168 (EX1046). (EX1002 ¶131; *see also* EX1049 (excerpt of RFC 793 webpage on IETF website, noting RFC 793 was updated by at least RFCs 1122 and 3168 along with RFCs 6093 (EX1047) and 6528 (EX1048)).) TCP implementations at the relevant time (i.e., before February 2010) also did not have to be fully compliant with any RFC document. (EX1002 ¶132.) Thus, a POSITA would not have been

¹⁴ The Board in the Google -845 IPR, explained that “given that RFC 793 predated the application for the ’995 patent by [at the time] 36 years, it is *not clear* that RFC 793 represents the ‘current’ version of TCP.” Google -845 IPR, Paper 16 at 18. The Board concluded that “on this record, we do not see how defining ‘TCP-variant’ with reference to ‘the current ‘TCP’” is particularly informative.” *Id.*

Petition for Post Grant Review
Patent No. 10,742,774

able to reasonably ascertain the bounds and scope of the “TCP” or “current TCP” features described by the ’774 patent with reasonable certainty, which is needed to frame a POSITA’s understanding as to the scope of the claimed “TCP-variant” features recited in claim 1 with reasonable certainty. (*Id.*) For instance, the ’774 patent does not provide any guidance as to whether RFC 793 and/or some other document(s) or source provides the requisite details needed to frame a POSITA’s understanding as to the components/characteristics of the “TCP” in a “TCP-variant connection” or “TCP-variant packet” at the relevant time of the ’774 patent. (*Id.* ¶133.) Indeed, none of these details are explained in the specification, claims, or file history. (*Id.*; *see generally* EX1004; EX1043; EX1045.)

Accordingly, for at least this reason, claim 1 is indefinite because when viewed in light of the specification and prosecution history, it does not inform a POSITA about the scope of the claimed “TCP-variant” features with reasonable certainty. (EX1002 ¶134.)

b) The Intrinsic Record Does Not Inform About the Scope of Variance Required to Encompass the Claimed “TCP-Variant” Features of Claim 1

In addition to the reasons above, the ’774 patent also does not provide clarity as to *what* variation or how *much* variation from the “[current] TCP” a packet or connection needs to include (or have to be within) such that the packet/connection

Petition for Post Grant Review
Patent No. 10,742,774

is considered to fall within the scope of the “TCP-variant” features of claim 1. (EX1002 ¶¶135–145.)

The Board considered the meaning of “variant” in “TCP-variant connection” in the Google -845 IPR involving the related ’995 patent (sharing a common specification with the ’774 patent). *See, e.g.*, Google -845 IPR, Paper 16 at 17–19 (October 8, 2020). While the Board applied the plain and ordinary meaning of “TCP-variant connection” as “a connection that varies from TCP” in that proceeding, *id.* at 18, the Board recognized that the specification provides little guidance as to [the] meaning” of a “TCP-variant connection.” *Id.* at 17; *see also id.* at 17–18 (noting that only the summary section mentions the TCP-variant term in a similar manner as the claims, “thereby providing no more guidance than the claims themselves.”). Thus, the Board preliminarily recognized the difficulty in ascertaining the proper scope of the “TCP-variant” terms in the ’774 patent family.

In any event, the Board’s and parties’ application of the plain meaning of the TCP-variant terms in that proceeding (and other pending IPRs addressing “TCP-variant” claims) does not detract from the indefiniteness positions presented here. *See, e.g., Flex Logix Tech. Inc. v. Konda*, PGR2019–00037, Paper 37 at 44 (Mar. 16, 2021) (finding challenged claims 1–7, 9–15, and 17–19 indefinite); *Flex Logix Tech. Inc. v. Konda*, PGR2019–00042, Paper 35 at 53 (Mar. 16, 2021) (simultaneously finding same challenged claims anticipated and obvious). While

Petition for Post Grant Review
Patent No. 10,742,774

anticipation/obviousness may be demonstrated by prior art that discloses one example of a “TCP-variant” packet/connection that includes a known TCP packet/connection implementation with additional structures or characteristics, it is another question altogether as to whether claim 1 (in light of the specification and file history) informs a POSITA as to the full scope of the claimed “TCP-variant” features beyond any one potential example. It does not.¹⁵

The ’774 patent specification lacks objective guidance regarding what connections and packets would qualify as a “variant” of a [current] TCP connection/packet. (EX1002 ¶136.) While “absolute or mathematical precision is

¹⁵ The prior art grounds in the now pending IPRs and prior art grounds in this PGR petition (*see infra* Section XI.B) that demonstrate unpatentability of PO’s “TCP-variant” claims do not contradict or detract from the indefiniteness positions against claim 1 here. Indefiniteness is not available for IPRs (*see, e.g.*, 35 U.S.C. § 311(b); *Samsung Elecs. Am. v. Prisia Eng’g Corp.*, 948 F.3d 1342, 1345 (Fed. Cir. 2020)) and the PGR grounds below demonstrate unpatentability consistent with the Board’s plain read of “TCP-variant” which must at least encompass protocols that “var[y] from TCP,” which *Eggert* discloses. (*See infra* Section XI.B; *see also supra* Section X.)

Petition for Post Grant Review
Patent No. 10,742,774

not required, it is not enough ... to identify ‘some standard for measuring the scope of the phrase.’” *Interval Licensing*, 766 F.3d at 1370–71 (quoting *Datamize, LLC v. Plumtree Software, Inc.*, 417 F.3d 1342, 1351 (Fed. Cir. 2005)). “The claims, when read in light of the specification and the prosecution history, must provide objective boundaries for those of skill in the art.” *Interval Licensing*, 766 F.3d at 1371.

There are simply no objective boundaries in the ’774 patent to inform a POSITA with reasonable certainty as to when a TCP connection is considered a “TCP-variant” versus a [current] TCP in context of the patent. (EX1002 ¶136.) The example in the specification that “an ITP header may be included in a footer of a protocol packet” could be considered an “extension and/or variant of the current TCP” (EX1001, 16:4–8) does not inform on the objective boundaries between a current TCP, an extension of the current TCP, or a variant of the current TCP, nor is there any such guidance elsewhere in the specification. (EX1002 ¶137.) Further, the “variant” term does not describe on its face what, where, or which components/characteristics are necessary or absent for a connection/packet to be a “TCP-variant connection”/“TCP-variant packet” in comparison to the other identified forms of TCP-based connections or packets. (*Id.* ¶138.) And even if that portion of the specification is considered, it does not provide any insight as to the scope of a “TCP-variant connection” as recited in claim 1. (*Id.*)

Petition for Post Grant Review
Patent No. 10,742,774

For instance, even if RFC 793 provides a baseline for the “TCP-variant” terms (which is unclear from the ’774 patent’s disclosure), the intrinsic record still does not inform a POSITA about the full scope of claim 1 with reasonable certainty. (EX1002 ¶139.) The claims, specification, and file history of the ’774 patent do not indicate which portions of RFC 793 a “TCP-variant” connection or packet must comply with or what aspects of a connection/packet causes such connection/packet to vary from the TCP enough to encompass the claimed “TCP-variant” connection/packet. As one example, a POSITA would not have been informed with reasonable certainty whether establishing a “TCP-variant connection” would require a three-way handshake, a feature that is required for establishing a TCP connection defined by RFC 793. (*See, e.g.*, EX1008, 31–32; EX1001, 14:63–15:10.) As another example, a POSITA would not have been informed with reasonable certainty whether a “TCP-variant connection” would require acknowledgment of sent packets and error recovery mechanisms (and which ones), which are features required by the TCP implementation defined in RFC 793. (*See, e.g.*, EX1008, 8 (“Reliability” section), 13–14 (section 2.6).) Therefore, the understanding of whether a packet/connection is a “TCP-variant” packet/connection may differ depending on what features and characteristics of a baseline TCP implementation are considered important, e.g., based on application-specific needs

Petition for Post Grant Review
Patent No. 10,742,774

and related design parameters—a subjective inquiry for which the ’774 patent provides no guidance or contours.¹⁶ (EX1002 ¶¶140–142.)

The claimed “TCP-variant [packet]/[connection]” limitation is similar to a term of degree that is subjective rather than objective, because the claims, specification, and file history do not provide any details concerning the required

¹⁶ The ambiguity in ascertaining the meaning of the claimed “TCP-variant” terms also exists to the extent PO alleges a POSITA would know how to identify a TCP packet/connection or a “variant” of such in context of the ’774 patent by using some software tool that presents a packet/connection’s attributes/characteristics. First, the ’774 patent provides no disclosure to support such a position. Second, even if considered, such a tool would not make the ultimate determination of whether a packet/connection is a “TCP-variant,” but rather would be left to the subjective view of a POSITA viewing such attributes/characteristics. The ultimate guidance needed to make that determination must come from the ’774 patent (and its intrinsic record). As explained, there are no disclosures in the ’774 patent to guide a POSITA in determining with reasonable certainty what attributes, characteristics, or any other features are needed to exist (or not exist) to identify a packet/communication as “TCP-variant.” (EX1002 ¶142 n.8.)

Petition for Post Grant Review
Patent No. 10,742,774

variance from a TCP packet/connection that would inform a POSITA as to the scope of the “TCP-variant” features with reasonable certainty. *Interval Licensing*, 766 F.3d at 1371; *UUSI, LLC v. United States*, 131 Fed. Cl. 244, 259 (2017). “[C]laims having terms of degree will fail of indefiniteness unless they ‘provide objective boundaries for those of skill in the art’ when read in light of the specification and prosecution history.” *Liberty Ammunition, Inc. v. United States*, 835 F.3d 1388, 1396 (Fed. Cir. 2016) (quoting *Interval Licensing*, 766 F.3d at 1370–71.) (EX1002 ¶143.)

Indeed, the term “variant” lacks a universal meaning, which further confounds the “TCP-*variant*” limitations of claim 1. (EX1002 ¶144.) Put another way, the ’774 patent’s disclosure does not inform a POSITA regarding the minimum or the maximum bounds of the word “variant” in the TCP-variant terms. For example, the court in *Collectis S.A. v. Precision Biosciences, Inc.* held that the term “variant of the wild-type monomer” was indefinite because nothing in the intrinsic record provided any indication as to scope of the term “variant” as it would have been used by a POSITA at the time of the alleged invention—i.e., there was no way to provide a clear delineation between when a monomer is variant or not. *See* 937 F. Supp. 2d 474, 483 (D. Del. 2013). Likewise, here, the intrinsic record of the ’774 patent fails to indicate the scope of “variant” as it would have been used to delineate

between a *TCP-variant* packet/connection and a *TCP* packet/connection. (EX1002 ¶144.)

Accordingly, for at least this additional reason, claim 1 does not inform a POSITA about the scope of the claimed “TCP-variant” features with reasonable certainty and thus is indefinite. (EX1002 ¶145.)

c) Other Claims of the '774 Patent and Claims From the '774 Patent's Parent Applications Further Demonstrate the Ambiguity Introduced by the “TCP-Variant” Features

The “TCP-variant” features were first introduced into the '774 patent family through claims in the '802 application (now the '995 patent). (EX1043, 63–64 (introducing claims with “TCP-variant connection” and “TCP-variant packet”).) From there, “TCP-variant” features were introduced in claims in the family of applications leading up to the '811 application, which issued as the '774 patent. (*See e.g., id.*, 75, 232; EX1045, 225; EX1004, 60–62, 136–37, 224.) As a result, the '774 patent now includes claims that introduce additional uncertainty as to how a POSITA could ascertain the scope of the claimed “TCP-variant” features recited in claim 1. (EX1002 ¶¶146–152.)

Specifically, some of the claims pursued during prosecution and now issued in the '774 patent demonstrate the ambiguity regarding the scope of what encompasses the “TCP-variant” features found in claim 1, such as:

Petition for Post Grant Review
Patent No. 10,742,774

- “the TCP-variant connection is *not a TCP connection*” (EX1001, 25:58–60 (claim 22));
- “the TCP-variant connection is *not a standard TCP connection*” (*id.*, 25:61–63 (claim 23));
- “the TCP-variant *includes another protocol, other than TCP, that includes a variation of TCP*” (*id.*, 27:26–27 (claim 40));
- “the TCP-variant *includes another protocol, without TCP in its name, that includes a variation of TCP*” (*id.*, 27:28–29 (claim 40));
- “the TCP-variant includes *one or more features of TCP, and one or more features not of TCP*” (*id.*, 27:30–31 (claim 40));
- “the TCP-variant packet includes *one or more features of a TCP packet, and one or more features not of a TCP packet*” (*id.*, 27:32–34 (claim 40)); and
- “the TCP-variant connection includes *one or more features of a TCP connection, and one or more features not of a TCP connection*” (*id.*, 27:35–37 (claim 40)).

(See also EX1004, 60–61.)

First, the features recited in the claims above are not described in the ’774 patent specification. (See *supra* Sections VIII, XI.A.1, XI.A.2.a–b; see generally EX1001; EX1002 ¶¶147–148.) Second, they raise more questions than answers as

Petition for Post Grant Review
Patent No. 10,742,774

to assisting a POSITA in ascertaining the scope of the claimed “TCP-variant” terms with reasonable certainty. (EX1002 ¶148.)

For example, the ’774 patent claims, file history, and specification do not provide details that would inform a POSITA with reasonable certainty as to how a “TCP-variant connection” can be a variant to a TCP connection but also “not [be] a TCP connection” as set forth in claim 22. (*Id.* ¶149.) Moreover, the interplay between claims 22 and 23 raises ambiguities as to the scope of claim 1’s “TCP-variant” features as there is no guidance in the ’774 patent that informs a POSITA with reasonable certainty as to what is meant by, and what is the difference between, a “standard” TCP connection and a non-standard TCP connection. (*Id.*) Similarly, with respect to claim 40, the ’774 patent does not provide any guidance to inform a POSITA with reasonable certainty as to what it means to “include[]” another protocol other than TCP that also includes a variation of TCP. (*Id.* ¶150.) Nor does the ’774 patent provide any guidance to inform a POSITA with reasonable certainty as to what it means to have (or not have) “TCP” in the “name” of a protocol while still being considered a “variation of TCP,” as alternatively set forth in claim 40. (*Id.*)

The “TCP-variant” terms also introduce an undefined outer boundary of the term “TCP-variant” where a connection/packet may vary from TCP so much that it would have been unclear whether it is still a “TCP-variant.” For example, to the

Petition for Post Grant Review
Patent No. 10,742,774

extent PO attempts to support the notion that “TCP-variant” somehow encompasses protocols that are not based on TCP (e.g., non-TCP connections or packets like connections and packets based on the IPX/SPX networking protocol), that interpretation has no support in the specification of the ’774 patent or any of its parent applications. (*Id.* ¶151.)¹⁷ Moreover, such a position and interpretation would conflict with the ’774 patent’s express recitation of “non-TCP” features in other claims. (*Id.* ¶151; EX1001, 29:10–23 (claim 42)); *Helmsderfer v. Bobrick Washroom Equip., Inc.*, 527 F.3d 1379, 1382 (Fed. Cir. 2008) (“different claim terms are presumed to have different meanings”).

Accordingly, the ’774 patent’s other claims (and file history pursuing such claims) undermines the ability for claim 1 to inform a POSITA about the scope of the claimed “TCP-variant” features with reasonable certainty, and thus for this additional reason, claim 1 is indefinite. (EX1002 ¶152.)

¹⁷ The indefiniteness of claim 1 based on the unclear scope of the claimed “TCP-variant” terms is further demonstrated by PO’s infringement contentions, mapping a “TCP-variant” connection/packet to a UDP-based protocol connection/packet (which are not TCP at all). (EX1017, 2; EX1002 ¶151 n.9.)

B. Ground 2: The Combination of *Wookey* and *Eggert* Renders Obvious Claim 1¹⁸**1. Claim 1****a) An apparatus comprising:**

To the extent limiting, *Wookey* discloses the features of this preamble. (EX1002 ¶¶153–160.) For example, *Wookey* provides a system (“apparatus”) (Figure 1) including client machines and remote machines such as **client machine 10**, **remote machine 30’**, and **remote machine 30**. (EX1005 ¶¶[0020], [0136]; *see also id.* ¶¶[0002], [0016], [0196].)

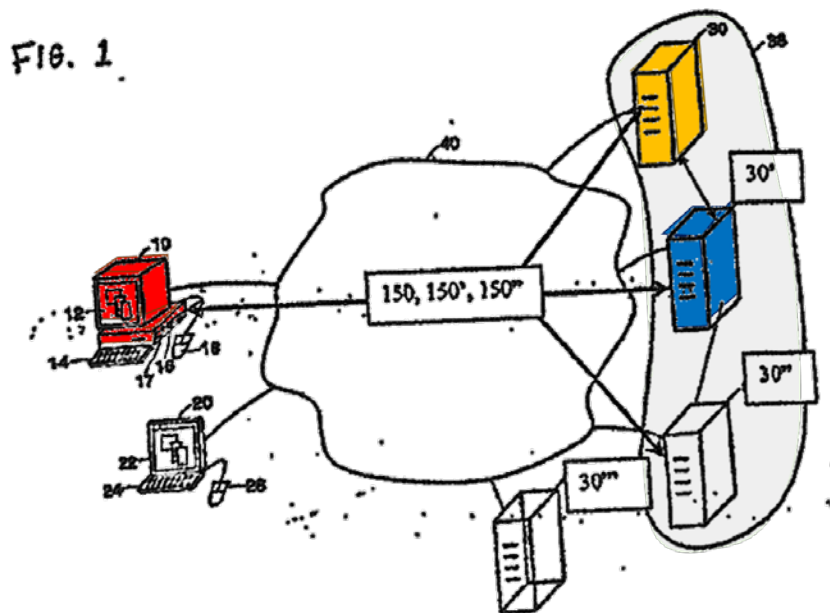
Figures 1, 2C, and 3D below show an exemplary system in which a **client machine 10**, **remote machine 30**, and **remote machine 30’**, among other components, are employed to permit a **client machine 10** access to a computer resource.¹⁹ (*E.g.*, EX1005 ¶¶[0016], [0020], [0136], [0196]; EX1002 ¶¶155–159,

¹⁸ Other references discussed herein are provided to show the state of the art at the time of the alleged invention of the ’774 patent.

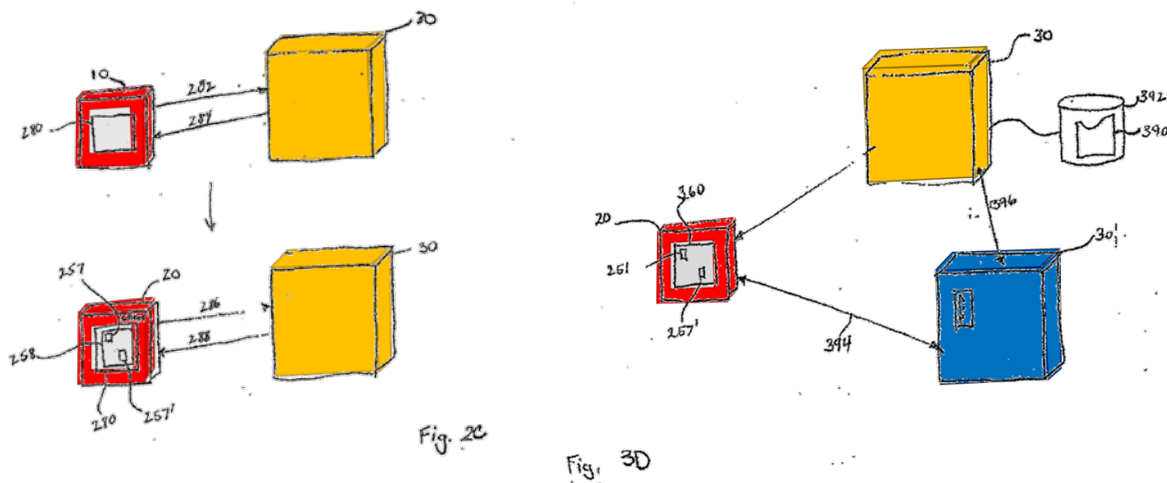
¹⁹ *Wookey*’s disclosures (including its figures) include certain typographical errors when referencing its client machines (referring to them interchangeably as 10, 10’ or 20). (EX1002 ¶159.) Moreover, in context of *Wookey*’s disclosures, any teachings regarding one client machine 10/10’/20 in *Wookey* apply equally to the other client machines. (*Id.*)

Petition for Post Grant Review
 Patent No. 10,742,774

74–83 (*Wookey* overview).). *Wookey* discloses that the client and remote machines are “typical computers” (EX1005 ¶[0021], FIGS. 1A–1B) including personal computers and wireless devices (*id.* ¶[0171]), such as mobile telephone or other portable telecommunication devices (*id.* ¶¶[0172]–[0173]).



Petition for Post Grant Review
Patent No. 10,742,774



(EX1005, FIGS. 1 (annotated), 2C (annotated), 3D (annotated) EX1002 ¶158.)

Thus, *Wookey* discloses an apparatus, as claimed. (EX1002 ¶160; *infra* Sections IX.B.1.b–f.)

- b) **a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for:**

Wookey discloses these limitations. (EX1002 ¶¶161–167.) For example, as noted above, *Wookey* discloses that its system includes **client machine 10** which is a computer (EX1005 ¶[0160]) that includes, as shown in Figures 1A and 1B (below), a main processor 102 (“one or more processors”) and two memory units—a main memory unit 104 and a cache memory 140 (individually or collectively, the claimed “non-transitory memory storing instructions”) (*id.* ¶¶[0161]–[0165]). (*See also supra* Section IX.B.1.a.) The main processor 102 is in communication with the main

Petition for Post Grant Review
Patent No. 10,742,774

memory unit 104 (*id.* ¶[0163]) and the cache memory 140 (*id.* ¶[0165]), e.g., via system bus 120. (*See also id.* ¶[0162], FIGS. 1A–B.)

FIG. 1A

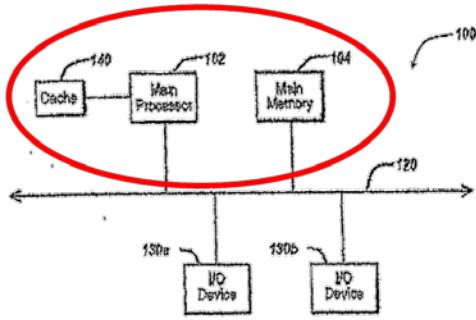
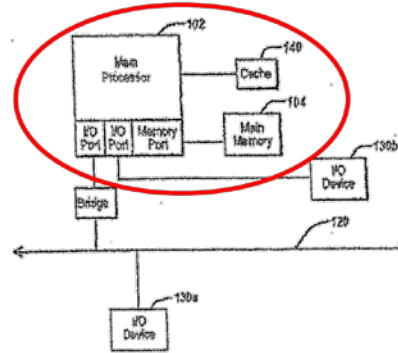


FIG. 1B



(*Id.*, FIGS. 1A–1B (annotated); EX1002 ¶162.)

In particular, figures 1A and 1B above depict typical computer architectures for both client and remote machines. (EX1005 ¶[0021].) Each computer 100, as shown above, includes a central processing unit (CPU) 102, main memory unit 104, and cache memory 140. (EX1005 ¶¶[0161], [0163].)²⁰ The CPU 102 “responds to and processes *instructions fetched from* the main memory unit 104.” (*Id.* ¶[0162]; *see also id.* ¶¶[0554], [0556], [0802], [1121], FIG. 33.) The main memory unit 104 can include one or more memory chips (e.g., SRAM) that store data accessed by

²⁰ *Wookey* uses the terms “Main Processor 102,” “microprocessor 102,” “processor 102,” and “central processing unit 102” interchangeably. (EX1005 ¶¶[0161], [0163]–[0164], FIGS. 1A–B; EX1002 ¶163 n.10.)

CPU 102. (*Id.* ¶[0163].) The cache memory 140 is a fast memory type that acts as a buffer between RAM and the CPU. (*Id.* ¶[0165].) As was known in the art at the time, cache memory stores both data and instructions accessible to the CPU, and thus *Wookey*'s cache 140 would necessarily store instructions accessible to the CPU. (EX1002 ¶¶163–165; EX1050, 1:15–39.) Accordingly, cache memory 140 and main memory unit 104 of **client machine 10** are each a non-transitory memory device and thus, individually or collectively, disclose the claimed non-transitory memory. (EX1002 ¶166.) As discussed below, the main processor 102 executes the instructions stored in the non-transitory memory to carry out the steps set forth in limitations 1.c–f. (*Id.*; *infra* Sections IX.B.1.c–f.)

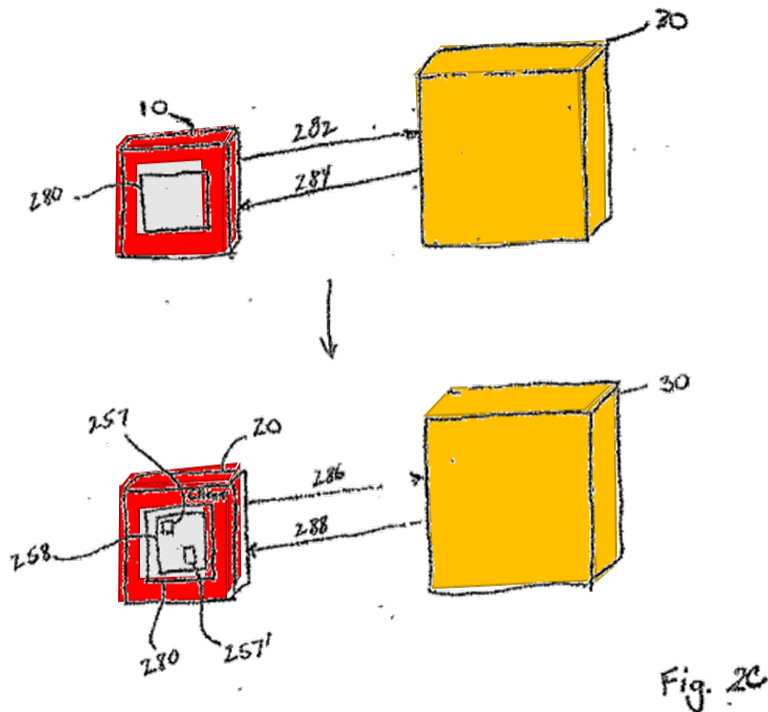
Therefore, *Wookey* discloses limitation 1.b. (EX1002 ¶167.)

- c) **receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established;**

The combination of *Wookey* and *Eggert* discloses and/or suggests this limitation. (EX1002 ¶¶168–216.) For example, as discussed below, the *Wookey-Eggert* combination discloses receiving, by **client machine 10** (“second node”) from **remote machine 30'** (“first node”), a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established. (*Id.* ¶168)

Petition for Post Grant Review
Patent No. 10,742,774

For example, **client machine 10**, as illustrated in Figure 2C below, executes a web browser application 280 (EX1005 ¶[0192]) that “transmits a request 282 to access a Uniform Resource Locator (URL) address corresponding to an HTML page residing on remote machine [30]” (*id.* ¶[0193]). (EX1002 ¶169.) The communication link between **client machine 10** and **remote machine 30** uses the transmission control protocol (TCP). (EX1005 ¶¶[0154]–[0156], [0174], [0731]–[0732].)

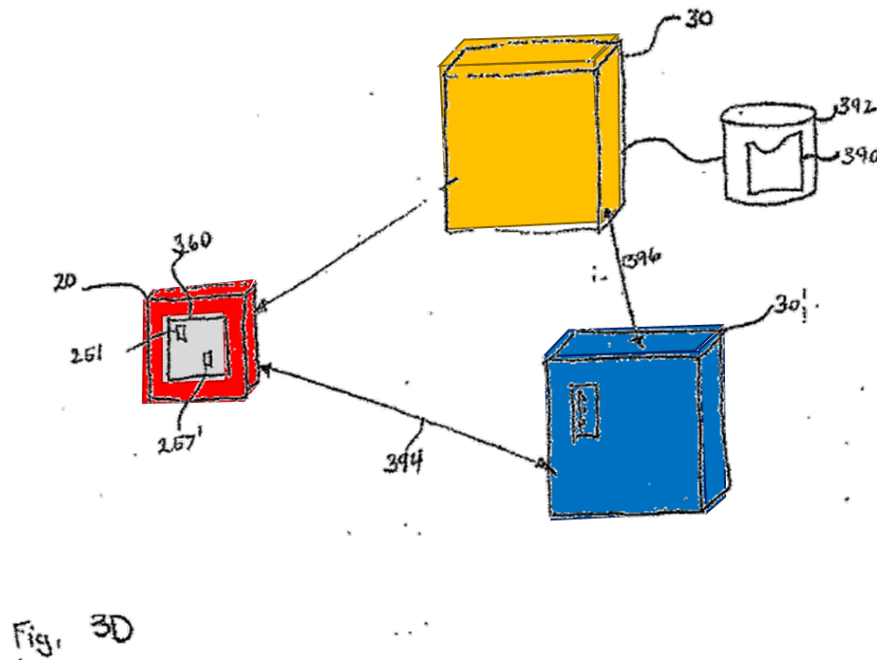


(*Id.*, FIG. 2C (annotated); EX1002 ¶169.)

Client machine 10 is then authenticated by **remote machine 30**. (EX1005 ¶[0193]; *see also id.* ¶¶[0196], [0207]–[0214].) After authenticating **client machine 10**, **remote machine 30** may “transmit[] to the client machine 10 an HTML page 288 that includes a Resource Neighborhood window 258 in which appears graphical icons 257, 257’ representing resources to which the client machine 10 has access.” (*Id.* ¶[0196].) The Resource Neighborhood window 258 displays graphical icons 257, 257’ representing resources to which **client machine 10** has access. (*Id.* ¶[0196]; *see also id.* ¶¶[0215]–[0216]; EX1002 ¶170.)

The HTML page 288 includes encoded URLs associated with icons (e.g., 257, 257’) on the HTML page 288, and by clicking on one of these icons 257, 257’, **client machine 10** can initiate a process for establishing a connection with **remote machine 30**’ (e.g., to access a desired resource on **remote machine 30**’). In particular, “[e]ach icon 257, 257’ is associated with an encoded URL that specifies: *the location of the resource ...; a launch command* associated with the resource; and *a template identifying how the results of accessing the resource should be displayed.*” (EX1005 ¶[0216].) *Wookey* also discloses that each encoded URL “contains the information necessary for the client to create a connection to the remote machine hosting the resource.” (*Id.*) Thus, when a user *clicks* an icon 257, 257’ corresponding to a resource from the displayed HTML page 288, **client machine 10** uses the encoded URL associated with the clicked icon to initiate the

process for establishing a second connection (e.g., connection 394 in Figure 3D below) with a second **remote machine 30'**. (*Id.*; *see also id.* ¶[0026]; EX1002 ¶171.)



(EX1005, FIG. 3D (annotated); *see also id.* ¶[0026]; EX1002 ¶¶171.)

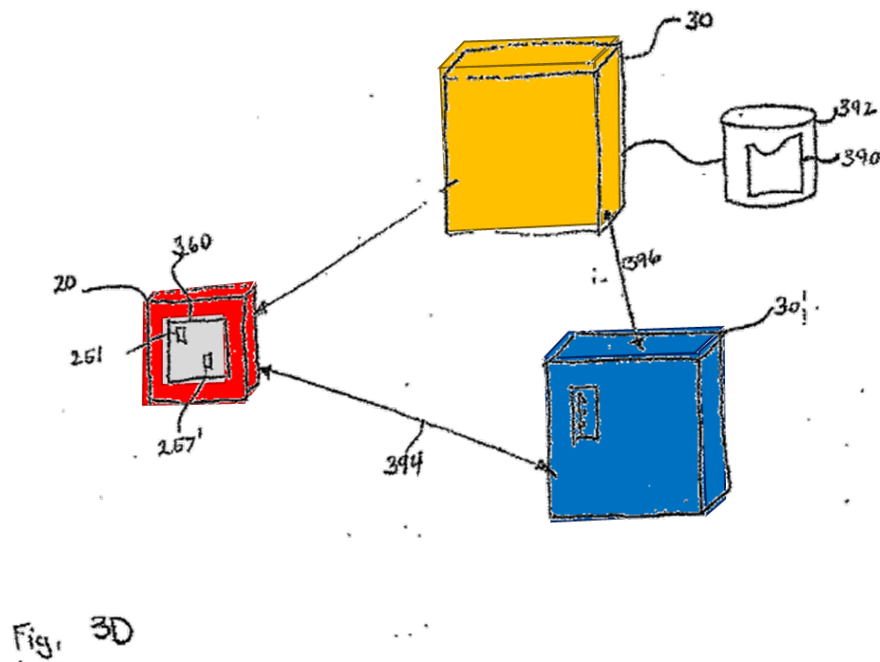
As discussed below, the process for **client machine 10** establishing a second connection with **remote machine 30'** includes receiving, by **client machine 10** (“second node”) from **remote machine 30'** (“first node”), a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established, as claimed, in the *Wookey-Eggert* combination. (*See also infra* Sections

Petition for Post Grant Review
Patent No. 10,742,774

IX.B.1.d–f (discussing additional steps in the process for establishing the connection).)

For example, regarding the connection between **client machine 10** and **remote machine 30**, *Wookey* provides that **client machine 10** includes an “HTTP client agent,” such as the web browser application 280, which “can use any type of protocol.” (EX1005 ¶[0159]; *see also id.* ¶[0155].) For example, connection 394 can be made using various protocols, including the Virtual Network Computing (VNC) protocol. (*Id.* ¶[0216]; EX1002 ¶173.) VNC is a graphical desktop-sharing system that allows a computer to remotely access and control another computer. (EX1002 ¶174; EX1022 ¶[0019].) VNC can run over various “industry standard transport protocols, such as TCP/IP, IPX/SPX, NetBEUI.” (EX1005 ¶[0225]; *see also id.* ¶¶[0155], [0215]–[0216]; EX1002 ¶175.)

Wookey further explains that **client machine 10** may use “a different type of protocol than the one used to send the request to the remote machine 30.” (EX1005 ¶[0732]; *see also id.* ¶¶[0738]–[0739], [0772].)



(*Id.*, FIG. 3D (annotated); *see also id.* ¶[0026]; EX1002 ¶176.)

Wookey also identifies desired characteristics associated at least with the second connection that a POSITA would have considered when choosing a protocol such as negotiating parameters related to the connection, reducing unintentional termination of sessions due to an imperfect connection, detecting and handling disconnections like when a mobile device enters an elevator, and ability to specify inactive time prior to connection termination. (EX1002 ¶177; EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153]; *see also id.* ¶¶[0738]–[0739] (disclosing that **remote machine 30'** determines which protocol is being used based on received connection request).)

Petition for Post Grant Review
Patent No. 10,742,774

Wookey further explains that machines **10** and **30'** “exchange a set of messages which negotiate the parameters under which communications will occur.” (EX1005 ¶[0774].) Once negotiations are complete, **client machine 10** and **remote machine 30'** “are able to communicate as necessary.” (*Id.*; see also *id.* ¶¶[0744], [0773].) Thus, **client machine 10** and **remote machine 30'** both send and receive packets when establishing a connection. (EX1002 ¶¶178–179.) And when using a TCP-based connection for the second connection, something that *Wookey* foresees as noted above, **client machine 10** would necessarily receive a TCP packet from **remote machine 30'** when establishing the second connection by performing a 3-way handshake. (*Id.*)

Moreover, when **client machine 10** uses one of the encoded URLs (e.g., the one pointing to **remote machine 30'**) associated with one of the icons on the HTML page 288, it results in **client machine 10** establishing a connection (e.g., connection 394) with **remote machine 30'** using a transport protocol. (EX1002 ¶180.)

While *Wookey* discloses that various transport protocols (e.g., TCP/IP, IPX/SPX) may be used to establish the second connection and desired characteristics of such a connection (EX1005 ¶[0225]), *Wookey* does not expressly provide details regarding establishing and maintaining the second connection or the types of negotiable parameters used to establish the second connection. (EX1002 ¶181.) However, it would have been obvious to a POSITA to consider and implement a

Petition for Post Grant Review
Patent No. 10,742,774

protocol (which may be “a different type of protocol than the one used to send the request to the remote machine 30” (EX1005 ¶[0732])) that would have met at least some of *Wookey*’s desired characteristics (e.g., such as those identified above) for the second connection between **client machine 10** and **remote machine 30’** and would have resulted in **client machine 10** “receiv[ing] ... a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established” like that recited in limitation 1.c. (EX1002 ¶181.)

Eggert describes such a protocol—a TCP-variant protocol using TCP-variant packets—that allows nodes to negotiate or modify timeout timers on a per-connection basis to provide a reliable and efficient network connection. Accordingly, as discussed below, based on the teachings of *Eggert*, the knowledge of a POSITA, and guidance from *Wookey*, it would have been obvious to configure *Wookey*’s **client machine 10** such that when **client machine 10** clicks on one of the icons 257, 257’ associated with one of the encoded URLs in the HTML page 288, **client machine 10**’s web browser application 280 operates in accordance with a protocol such as the one described by *Eggert* to initiate establishment of the connection between **client machine 10** and **remote machine 30’**, in order to “receiving, by a second node from a first node, a *transmission control protocol (TCP)-variant* packet in advance of a *TCP-variant* connection being established.” (EX1002 ¶182.)

(1) *Eggert*

Eggert, like *Wookey*, relates to establishing a reliable connection between two nodes. For instance, where a node may be mobile, the node may experience “changes [in] network attachment points based on [its] current location,” and thus *Eggert* discloses networking features in the same technical field as *Wookey*. (EX1006, 1–3; EX1002 ¶¶183–185; *infra* Section IX.B.1.c.2 (discussing similarities between *Wookey* and *Eggert*).)

Eggert describes a modification to the TCP protocol to “allow established TCP connections to survive periods of disconnection” (EX1006, Abstract). In particular, *Eggert* introduces a “TCP Abort Timeout Option” (ATO) that “allows conforming TCP implementations to negotiate individual, per-connection abort timeouts.” (*Id.*) Thus, as explained below, *Eggert* discloses a TCP-variant protocol that allows established TCP connections to survive periods of disconnection by negotiating abort timeouts. (*Id.*; EX1002 ¶186; *see also id.* ¶¶84–93 (*Eggert* overview).)

Eggert explains that its TCP-variant protocol solves the problem “where hosts are only intermittently connected to the Internet.” (EX1006, 1–3.) For example, a mobile host may “experience disconnected periods during which no network service is available” when “mobile hosts ... change network attachment points.” (*Id.* 2–3.) These disconnected periods may lead to the “established TCP connections” being

Petition for Post Grant Review
Patent No. 10,742,774

“abort[ed] during periods of disconnection.” (*Id.*) *Eggert* overcomes this problem in the art by allowing hosts to negotiate per-connection abort timeouts. (*Id.*) *Eggert*’s TCP-variant protocol “allow[s] mobile hosts to maintain TCP connections across disconnected periods that are longer than their system’s default abort timeout.” (*Id.*; EX1002 ¶187.)

Eggert further explains that if “[a] TCP implementation” on a device “does not support the TCP [ATO],” then the device “SHOULD silently ignore it.” (EX1006, 7.) *Eggert*, thereby, provides a new option to the traditional TCP implementation where devices configured according to *Eggert* vary from the “traditional TCP implementation” in terms of the ATO negotiation. (EX1002 ¶¶145–146.) And as discussed below, just like the ’774 patent, *Eggert*’s ATO is also incorporated in the TCP header and arranged in a similar format. (*Compare* EX1001, 15:59–67, FIG. 8 *with* EX1006, 3–5, FIG. 1; EX1002 ¶¶188–191.)

Thus, a connection using *Eggert*’s ATO would be a TCP-variant connection and at least the segments used for establishing such a TCP-variant connection (e.g., the segments using the TCP ATO) would be TCP-variant packets. (*See also supra* Section X; EX1002 ¶¶192–193.)

Eggert’s TCP-variant protocol uses the three-way handshake to establish a connection between two nodes on a network. (EX1006, 5; EX1002 ¶194.) A three-way handshake involves the exchange of synchronize (SYN) and acknowledge

Petition for Post Grant Review
Patent No. 10,742,774

(ACK) messages between the nodes to establish a connection—i.e., SYN, SYN-ACK, and ACK messages in a standard TCP implementation. (EX1002 ¶194; EX1008, 31–32, 34–37, *see also* EX1001, 14:63–15:10 ('774 patent disclosing the TCP three-way handshake).) The connection between the nodes is established at the completion of the three-way handshake. (EX1002 ¶194.)

As discussed below and exemplified in Figure 2 below, *Eggert* discloses the features of this limitation in two different ways such that that a first node receives a TCP-variant packet from a second node during the exchange of three messages (3-way handshake) in *Eggert*'s TCP-variant protocol for negotiating a per-connection abort timeout, in advance of the TCP-variant connection being established.²¹

²¹ In the combined *Wookey-Eggert* process/system discussed below (*e.g.*, *infra* Section IX.B.1.c.2), the combined process/system discloses the claim limitations (*e.g.*, limitations 1.d–g) under both embodiments shown in *Eggert* (*i.e.*, left and right side of Figure 2).

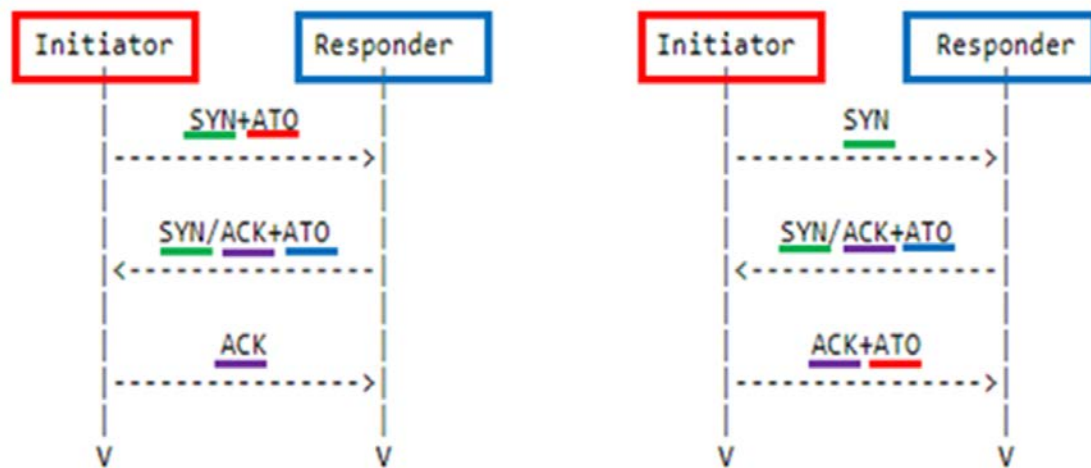


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

(EX1006, FIG. 2 (annotated); EX1002 ¶195.)

First, as shown on the left-hand side of Figure 2 (when the **initiator** initiates an abort timeout negotiation), *Eggert* teaches that the **initiator** (e.g., client node) sends the **responder** (e.g., server node) a “**SYN+ATO**” segment.²² (EX1006, FIG.

²² The '774 patent describes a “packet” in a manner consistent with what the TCP standard (RFC 793) describes as “segment” at that time. (*Compare* EX1001, 15:38–58, FIG. 8 *with* EX1008 (RFC 793), 15, FIG. 3 (showing the same TCP header in a TCP “packet” (EX1001) and TCP “segment” (EX1008)).) Thus, *Eggert*’s TCP “segment” is the same as the TCP “packet” as described and claimed in the '774 patent. (EX1002 ¶196 n.16.)

2; *see also id.*, 5–7.) The ATO in the segment indicates that this segment contains an abort timeout. (*Id.*, 3.) *Eggert* explains that when the **responder** (e.g., server node) accepts the **initiator's** offered abort timeout, it must echo the offered timeout value in the ATO it sends. (EX1006, 5–7.) That is, the **initiator** receives an “SYN/ACK+ATO” segment (“TCP-variant packet”) from the **responder**. (*Id.*; EX1002 ¶196.)

Second, as shown on the right-hand side of Figure 2 (when the **responder** initiates an abort timeout negotiation), *Eggert* teaches that the **initiator** (client node) receives an “SYN/ACK+ATO” segment (“TCP-variant packet”) from a **responder** (server node). (EX1006, 5–7; EX1002 ¶197.)

The received (SYN/ACK+ATO) segment in both scenarios (left and right side of Figure 2) is a TCP-variant packet because it contains an abort timeout option (ATO), which is an option not standard in the TCP protocol as defined by RFC 793. (EX1002 ¶198.) *See also* Google -845 IPR, Paper 16 at 20–21 (Board agreeing that *Eggert* discloses a TCP-variant packet for similar reasons).

Indeed, *Eggert's* ATO shown in Figure 1 uses the same TCP structure as the idle time period (ITP) field illustrated in Figure 8 of the '774 patent. For example, like the '774 patent, *Eggert's* ATO is also incorporated in the TCP header and arranged in a similar format. (*Compare* EX1001, 15:59–67, FIG. 8 *with* EX1006, 3–5, FIG. 1.) As shown below, both formats use a KIND sub-field to indicate the

Petition for Post Grant Review
Patent No. 10,742,774

new option, a TCP option length sub-field, and a data portion (YELLOW) of the TCP option containing the timeout value. (*Compare* EX1001, FIG. 8 *with* EX1006, FIG. 1.)

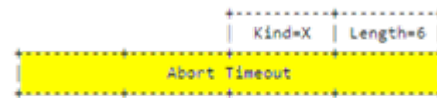
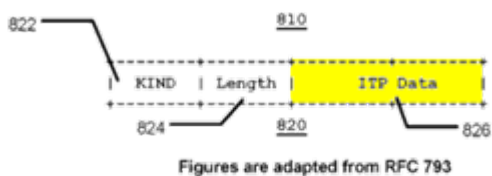


Figure 1: TCP Abort Timeout Option

(EX1001, FIG. 8 (annotated) (left); EX1006, FIG. 1 (annotated) (right); EX1002 ¶199.) Thus, *Eggert* uses the same format for the ATO as the embodiment disclosed in the '774 patent for a TCP-variant packet. (EX1002 ¶200.) Accordingly, as explained above, a connection using *Eggert*'s ATO would be a TCP-variant connection and at least the segments used for establishing such a TCP-variant connection (e.g., the segments using the ATO) would be TCP-variant packets. (*Id.*; *see also supra* Section X.)

The understanding that *Eggert*'s "SYN/ACK+ATO" segment is a TCP-variant packet in the context of the '774 patent is also consistent with PO's proposed construction of "variant" in a related matter. (*See, e.g.*, EX1018, 60 (Exhibit C) (PO offering definition of variant as one that "manifest[s] variety, deviation, or disagreement, [or] var[ies] slightly from the standard.")) *Eggert* describes a TCP-variant packet under PO's interpretation because the packets exchanged during the

three-way handshake in *Eggert* vary, as described above, from those exchanged in the TCP standard.

Eggert thus discloses the features of “receiving, by a second node from a first node, a *transmission control protocol (TCP)-variant* packet in advance of a *TCP-variant* connection being established,” like that claimed. (EX1002 ¶201.)

(2) Reasons to Combine

Based on *Wookey*’s and *Eggert*’s disclosures, a POSITA’s knowledge, and the discussions above, it would have been obvious to configure and implement *Wookey*’s **client machine 10** to use a TCP-variant protocol like the one disclosed by *Eggert* when establishing (and using) the second connection between **client machine 10** and **remote machine 30**’ concerning claim limitations 1.c–1.f. Such an implementation would have had many advantages and been a predictable and straightforward result achieved by combining well-known technologies using known methods. (EX1002 ¶202.) *See KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 416–18 (2007).

Eggert and *Wookey* disclose features in a similar technological field. (EX1002 ¶203.) For example, both *Wookey* and *Eggert* are concerned about maintaining connections in situations where at least one node (e.g., a mobile node) in the connection may lose connectivity due to its movement. (*Id.*) *Wookey* discloses that “client machine 10 may traverse network segments or network access

Petition for Post Grant Review
Patent No. 10,742,774

points that cause changes in the network address ... or causes the client machine 10 to disconnect.” (EX1005 ¶¶[0581]; *see also id.*, ¶¶[1135]–[1136] (discussing the need for connections to manage unintentional disconnection between the client and remote machines), [1153] (discussing an unintentional disconnection when **client machine 10** enters an elevator).) Similarly, *Eggert* discloses that “[l]engthening abort timeouts allows established TCP connections to survive periods of disconnection” as “[s]ome hosts are only intermittently connected to the Internet.” (EX1006, 1–3.) Like *Wookey*, *Eggert* discloses that “[o]ne example [of such unintentional disconnections] is mobile hosts that change network attachment points based on current location” and “[i]n between connected periods, mobile hosts may experience disconnected periods during which no network service is available.” (*Id.*, 2–3.) Thus, both *Wookey* and *Eggert* are directed to establishing reliable connections between two nodes, where the nodes may be mobile nodes that experience an unintentional disconnection, so a POSITA would have had reason to consider *Eggert* when contemplating and implementing *Wookey*’s teachings. (EX1002 ¶¶204–206.)

Moreover, as explained above, *Wookey* describes providing access to icons 257, 257’ on an HTML page 288 that, when used by **client machine 10**, causes **client machine 10** to establish a connection between **client machine 10** and **remote machine 30’** using a protocol that can be different from the one used to connect

Petition for Post Grant Review
Patent No. 10,742,774

client machine 10 and **remote machine 30**. *Wookey* further discloses the types of characteristics a POSITA would have considered concerning the protocol that can be used for the second connection, such as negotiating parameters related to the connection, reducing unintentional termination of sessions due to an imperfect connection, detecting and handling disconnections like when a mobile device enters an elevator, and ability to specify a permissible inactive time before connection termination. (EX1005 ¶¶[0581], [0721], [0751], [1134]–[1136], [1153]; *see supra* Section IX.B.1.c.) A POSITA would have been thus motivated by *Wookey*’s disclosures and suggestions to consider protocols that would provide these desired features for the connection with **remote machine 30’**, including the features associated with the TCP-variant connection described by *Eggert*. Such a POSITA would have been motivated to incorporate a protocol like *Eggert*’s TCP-variant protocol for use by **client machine 10** for establishing a connection with **remote machine 30’**. (EX1002 ¶¶207–208.)

Using the TCP variant protocol with an ATO option as described in *Eggert* would have provided **client machine 10** with benefits including an improved, reliable connection to **remote machine 30’** with the versatility of negotiated timeout parameters consistent with *Wookey*’s desired characteristics for the second connection. *See KSR*, 550 U.S. at 416. Indeed, by using a TCP variant protocol with an ATO option (or similar feature as described by *Eggert*), *Wookey*’s method could

Petition for Post Grant Review
Patent No. 10,742,774

“negotiate individual, per-connection abort timeouts” (EX1006, 1) “to survive periods of disconnection” (*id.*) over a reliable connection. (EX1002 ¶209.) Thus, *Wookey*’s method for “providing a client with a reliable connection to a host service” (EX1005 ¶[0077]) would have been improved by allowing “[l]ong abort timeout values” so that the “hosts ... tolerate extended periods of [temporary] disconnection,” as provided by *Eggert* (EX1006, 5). (EX1002 ¶209.) Moreover, configuring *Wookey* to utilize a TCP-variant protocol based on *Eggert*’s teachings would have been both a predictable and straightforward implementation. (EX1002 ¶210; EX1005 ¶[0216].); *see KSR*, 550 U.S. at 416.

Additionally, *Eggert*’s protocol aims to avoid or mitigate interruptions caused by intermittent disconnections between the nodes, a problem recognized by *Wookey*. (EX1002 ¶211; EX1006, 1–3; EX1005 ¶[0903].) Thus, a POSITA would have been motivated to consider *Eggert*’s features when considering how to implement *Wookey*’s process/system and modify *Wookey* as discussed. Indeed, *Eggert*’s TCP variant protocol, which allows the nodes to negotiate an abort timeout, would have improved *Wookey*’s process just as described in *Eggert*. (EX1002 ¶211.) Accordingly, such a combination would have yielded the predictable result of providing a network communication exchange that minimizes disruptions caused by timeout issues. (*Id.*); *see KSR*, 550 U.S. at 417.

Petition for Post Grant Review
Patent No. 10,742,774

A POSITA would have further recognized that the *Wookey-Eggert* combination would have involved the use of known technologies (e.g., aspects of similar protocols) and design concepts and processes to obtain the foreseeable result of a reliable connection between **client machine 10** and **remote machine 30'**. (EX1002 ¶212). The above-modification would have involved substituting features from one reliable protocol from a finite number of available alternative reliable communication protocols, such as that described by *Eggert*. (*Id.*) Thus, a POSITA would have had a reasonable expectation of success in the above-modification. (*Id.*) *See Pfizer, Inc. v. Apotex, Inc.*, 480 F.3d 1348, 1364 (Fed. Cir. 2007) (“only a reasonable expectation of success, not a guarantee, is needed” in an obviousness analysis).²³

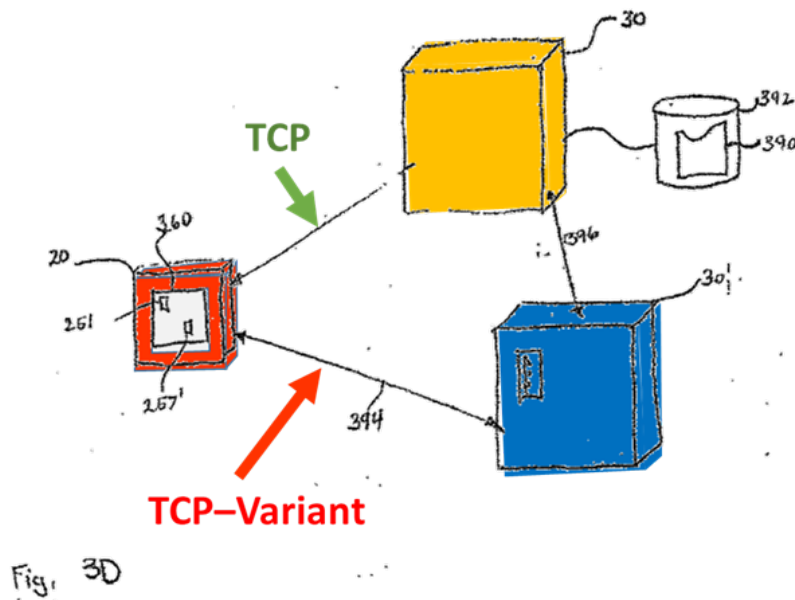
A POSITA would thus have found configuring *Wookey's* system/process such that the encoded URLs in HTML page 288 or page 288 including the encoded URLs (associated with the icons 257, 257'), when used by **client machine 10**, would cause

²³ There is no evidence of nonobviousness at this time that would overcome the overwhelming evidence of obviousness as presented herein. Petitioner reserves the right to address any such evidence/argument should PO later present any in this proceeding.

client machine 10 to utilize a TCP-variant protocol like *Eggert's* TCP-variant protocol a foreseeable and straightforward implementation. (EX1002 ¶213); see *KSR*, 550 U.S. at 416. For example, as explained, *Wookey* discloses that “*the URL includes a file, or a reference to a file, that contains the information necessary for the client to create a connection to the remote machine hosting the resource.*” (EX1005 ¶[0216].) In light of the reasons discussed above, a POSITA would have been motivated to configure *Wookey's* encoded URL such that it includes information causing **client machine 10** to establish a connection with **remote machine 30'** using the TCP-variant protocol (like that described by *Eggert*). (EX1002 ¶214.) This implementation could have been achieved by combining well-known technologies using known methods, such as known network design concepts and technologies described by *Wookey* and *Eggert* and known in the art at the time. (*Id.*)

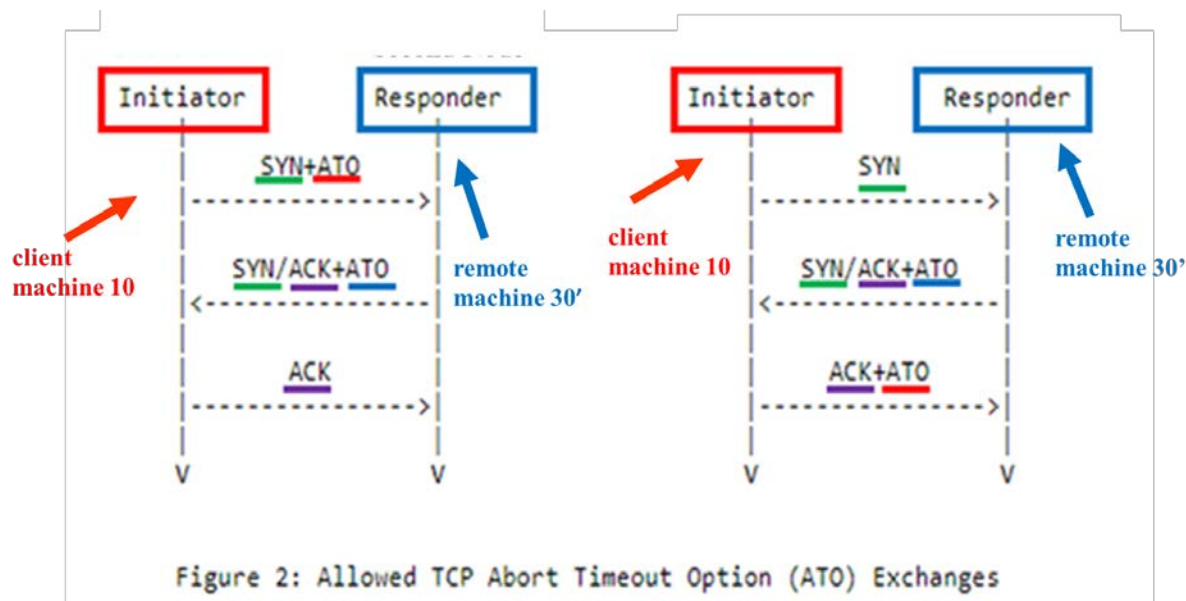
Indeed, a POSITA would have been skilled and would have had the knowledge to configure *Wookey's* system and method to implement a TCP-variant packet and protocol in various ways, while taking into account any known programming, design, and other related concepts, limitations, benefits, and the like to ensure the resulting combination operated properly and as intended. For example, a POSITA would have been motivated based on such disclosures to configure

Wookey's system and process consistent with the exemplary annotated figures from Wookey and Eggert below.²⁴



(EX1005, FIG. 3D (annotated); EX1002 ¶215.)

²⁴ The configurations discussed below are exemplary and not limiting. (EX1002 ¶215 n.17.)



(EX1006, FIG. 2 (annotated); EX1002 ¶215.)

Accordingly, the *Wookey-Eggert* combination as discussed above discloses and/or suggests that when during the process of establishing a connection between **client machine 10** and **remote machine 30'**, **client machine 10** (“second node”) would have performed a three-way handshake with **remote machine 30'** (“first node”), which would have included **client machine 10** receiving, from **remote machine 30'**, a TCP-variant packet in advance of a TCP-variant connection being established (which would be established at the completion of the three-way handshake). (See *supra* Section XI.B.1.c.2 (discussing the *Wookey-Eggert* combination); EX1002 ¶216.) Therefore, the *Wookey-Eggert* combination discloses and/or suggests limitation 1.c. (EX1002 ¶216.)

d) detecting a time period parameter field in the TCP-variant packet;

The *Wookey-Eggert* combination discloses and/or suggests this limitation. (EX1002 ¶¶217–228.) As discussed above for claim limitation 1.c, the combined *Wookey-Eggert* system/process would have involved negotiating the abort timeout option included in the received “SYN/ACK+ATO” segment (“TCP-variant packet”) at **client machine 10** for establishing a TCP-variant connection with **remote machine 30**. (*Supra* Section IX.B.1.c.)

Eggert explains that this negotiation includes the **initiator** detecting the ATO field (“time period parameter field”) in the “SYN/ACK+ATO” segment (“TCP-variant packet”).

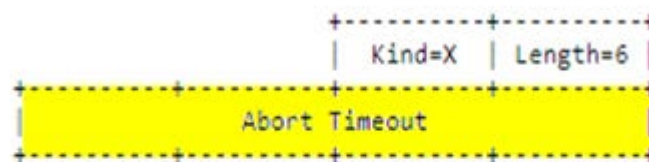


Figure 1: TCP Abort Timeout Option

(EX1006, FIG. 1 (annotated); EX1002 ¶¶218–220.)

First, regarding the left-hand side of Figure 2, *Eggert* provides that the **initiator** must accept the **responder**’s returned abort timeout value because the **initiator** proposed the abort time in the initial “SYN+ATO” segment it sent to the

Petition for Post Grant Review
Patent No. 10,742,774

responder. (EX1006, 5–7 (explaining that “[t]he host that initially proposed the Abort Timeout Option analyzes the next segment it receives from its peer” and if the next segment contains the ATO, “the connection MUST use the abort timeout contained inside the Abort Timeout Option.”); EX1002 ¶221.)

Second, regarding the right-hand side of Figure 2, the **initiator** may either accept or shorten the offered abort timeout from the **responder**. (EX1002 ¶222.)

Thus, for either side of Figure 2, the **initiator** (e.g., client node) must first detect the ATO parameter field containing the ATO value before it can accept or shorten the value. (EX1006, 5.) In this way, *Eggert* discloses detecting an ATO field in the TCP-variant packet. (EX1002 ¶223.)

The ATO field represents a time period parameter field used to convey an abort timeout value between two hosts. (EX1006, 1–7, FIG. 1.) The ATO field contains the abort timeout value, which defines a “time period,” as that claimed. (EX1006, 2–5, FIG. 1.) For example, the abort timeout value in the ATO field “is the desired abort timeout of the connection, specified in seconds.” (*Id.*, 3, FIG. 1; *see also id.*, Abstract.) *Eggert* further explains that by “[l]engthening [the] abort timeout” value that the “established TCP connections” can “survive periods of disconnection.” (*Id.*, Abstract.) The abort timeout value defines a time period because it is associated with a period where no packet is communicated in the

Petition for Post Grant Review
Patent No. 10,742,774

connection to keep the connection active. (EX1002 ¶¶224–226; *see also* EX1006, Abstract; *see also infra* Section IX.B.1.e.)

Accordingly, for reasons similar to those explained above, the *Wookey-Eggert* combined process would have been further configured to include the process of detecting, by **client machine 10**, an ATO field (“time period parameter field”) in the “SYN/ACK+ATO” segment (“TCP-variant packet”) received from **remote machine 30’**. (*See supra* Section XI.B.1.c.2 (discussing the *Wookey-Eggert* combination); EX1002 ¶227.) Accordingly, the *Wookey-Eggert* combination discloses and/or suggests limitation 1.d. (EX1002 ¶228.)

- e) **identifying metadata in the time period parameter field for a time period, during which no packet is received from the first node in a data stream of the TCP-variant connection and processed by the second node to keep the TCP-variant connection active; and**

The *Wookey-Eggert* combination discloses and/or suggests this limitation. (EX1002 ¶¶229–246.) As discussed below, *Eggert* discloses identifying an abort timeout value (“metadata”) in the abort timeout field (“time period parameter field”) for an abort timeout (“time period”), during which no packet is received from **remote machine 30’** (“first node”) in a data stream of the TCP-variant connection and processed by **client machine 10** (“second node”) to keep the TCP-variant connection active. (*Id.*; *see also supra* Section IX.B.1.d.)

Petition for Post Grant Review
Patent No. 10,742,774

Eggert discloses that the **initiator** (client node) “identif[ies] metadata in the time period parameter field for a time period,” as claimed. For example, as discussed above for claim limitation 1.d, *Eggert* explains that the **initiator** accepts or shortens the offered abort timeout value (“metadata”) specifying an abort timeout (“time period”) contained within the “SYN/ACK+ATO” segment (“TCP-variant packet”). (*Supra* Section XI.B.1.d; EX1006, 5–7.) Thus, the **initiator** must first identify the abort timeout value before it can either accept or shorten the offered abort timeout. (EX1006, 5–7; EX1002 ¶230.) Consequently, *Eggert* discloses that the **initiator** processes the received “SYN/ACK+ATO” segment from the **responder** according to an established format (EX1006, FIG. 1) to identify which bits of the option represents the proposed abort timeout duration value. (*Id.*, 3–5; *see also* EX1002 ¶231.)

Eggert explains that the “Abort Timeout” is “specified in seconds” (EX1006, 3) and is represented by a “32-bit value” (*id.*, 9), and thus the bits that identify the ATO field as associated with an abort timeout (KIND, LENGTH) and the bits representing the abort timeout value each constitute “metadata” for the proposed or responsive abort timeout (“time period”). (EX1002 ¶232.) Further, as discussed above for claim limitation 1.c, the format for representing the ATO in Figure 1 of *Eggert* is similar to the ITP data field 826 in Figure 8 of the ’774 patent. (*Id.*; *supra* Section XI.B.1.c.)

Petition for Post Grant Review
Patent No. 10,742,774

The manner as to how *Eggert* discloses identifying such values is consistent with the '774 patent's discussions. For example, the '774 patent explains that the metadata can be a "duration of time" (EX1001, 16:8–16, 22:5–10), which is what the ATO value in *Eggert* contains (EX1006, 3–5, FIG. 1; EX1002 ¶¶233–234.) Moreover, the understanding that *Eggert*'s descriptions regarding identifying values, such as the abort timeout value, discloses identifying metadata like that claimed is consistent with PO's view of the same claimed features. (EX1017, 9 (PO alleging "identifying metadata (e.g., a value of the idle time out parameter field, etc.) in the time period parameter field (e.g., timeout parameter field, etc.) for a time period"), ("the metadata includes a value in seconds").)

Eggert teaches that the abort timeout value corresponds to a "time period, during which no packet is received from the first node in a data stream of the TCP-variant connection ... to keep the TCP-variant connection active," as claimed. (EX1002 ¶¶234–235.) For example, *Eggert* explains that the nodes use the abort timeout value to track periods during which no packet is received in a data stream of the TCP-variant connection to keep the connection active. (EX1006, 1–7; *see also id.*, 9.) A POSITA would have recognized that the packet that must be "received ... to keep the ... connection active" in *Eggert* is an ACK that a node receives from the other node in response to a previous sent segment (where the ACK is received in the data stream through which the segment was sent). (EX1002 ¶¶235–236.) Indeed,

Petition for Post Grant Review
Patent No. 10,742,774

Eggert's ATO relies upon "[t]he TCP specification [1] [which] includes a 'user timeout' that defines the maximum amount of time that segments may remain *unacknowledged* before TCP will abort the connection. If a disconnection lasts longer than the user timeout, the TCP connection will abort." (EX1006, 3).)

Moreover, *Eggert* teaches that the abort timeout value corresponds to a "time period, during which no packet is received from the first node in a data stream of the TCP-variant connection *and processed by the second node* to keep the TCP-variant connection active," as claimed. (EX1002 ¶¶237–239.) For example, as noted above, *Eggert*'s implementation is based on the TCP specification set forth in RFC 793 (EX1008), with the added ATO which "allows conforming TCP implementations to negotiate individual, per-connection abort timeouts." (EX1006, 1; *see also id.*, 3 (referencing RFC 793, which is reference "[1]," when discussing communication between two nodes), 5 (describing that in its operation, during certain other states, normal timeouts are used in accordance with RFC 793), 9 (discussing the length of abort timeouts as specified in RFC 793), 15 (identifying reference [1] as RFC 793).). Thus, a POSITA would have understood that in *Eggert*, a received packet such as an ACK packet would have been processed in accordance with the disclosures of RFC 793.

In light of this understanding, it would have been obvious to a POSITA that in *Eggert*, if an ACK packet is received during the user timeout, the receiving node

Petition for Post Grant Review
Patent No. 10,742,774

would have performed processing on the received ACK packet to link the acknowledgment in the ACK packet with a previously sent data packet to keep the TCP-variant connection active. (EX1002 ¶240.) For example, the sequence number in the received ACK packet would be compared with the sequence number in the sent data segment to ensure the acknowledgment is related to the previously sent packet, to keep the TCP-variant connection active. (*Id.*; EX1008, 28 (section 3.3).) Thus, *Eggert* suggests that a packet needs to be received and processed by the client node to keep the TCP-variant connection active.

Accordingly, *Eggert*'s description of the abort timeout as the duration of time during which a node can wait to receive and process an ACK is a period "during which no packet is received from the first node in a data stream of the TCP-variant connection and processed by the second node to keep the TCP-variant connection active." (EX1002 ¶241.)

Disconnection

The combined *Wookey-Eggert* process/system discloses or suggests that the abort timeout value in the ATO field corresponds to a "time period, during which no packet is received from the first node in a data stream of the TCP-variant connection and processed by the second node to keep the TCP-variant connection active," as claimed, in an **additional** way where a disconnection occurs between **client**

Petition for Post Grant Review
Patent No. 10,742,774

machine 10 and **remote machine 30**'.²⁵ (EX1005 ¶¶[0581], [1135]–[1136], [1153]; EX1006, 1–3; *see also supra* Section XI.B.1.c.2 (noting that both *Wookey* and *Eggert* aim to maintain connection between nodes through unintentional disconnection periods); EX1002 ¶242.)

For example, when an interruption in communication occurs such as a disconnection in the network service (as contemplated by *Wookey* and *Eggert*) or in a physical network medium (as described in the '774 patent) (EX1001, 2:16–19, 2:33–36, 13:5–14), packets from one node do not reach the other node in either direction. (EX1002 ¶243.) In this situation contemplated by the '774 patent, no packets are received from the **responder** in the data stream of the TCP-variant connection and processed by the **initiator** to keep the connection active because no packet can get through the interruption or the physical medium's disconnection. (EX1002 ¶243; EX1001, 13:5–14, 2:33–36, FIG. 7 (annotated below in yellow showing no communication between the nodes during a disconnection).) Indeed,

²⁵ This is consistent with how the '774 patent describes such features. (EX1001, 13:5–14, 2:33–36 (describing that the idle time period may be linked to the physical disconnection of a network medium or simply a dead connection); EX1002 ¶242 n. 18.)

Petition for Post Grant Review
Patent No. 10,742,774

neither node would have received any packets from the other node during such a disconnection period. Therefore, neither node would have attempted to send a responsive acknowledgment packet during the period. (EX1002 ¶244.) This is the type of situation where, as discussed above, *Eggert* aims to maintain the connection, e.g., maintain the connection even when no packet is received by the **initiator** from the **responder** in the data stream of the TCP-variant connection during the negotiated abort timeout period. And since no packet would have been received by the **initiator**, no packet would have been processed by the **initiator** (e.g., there would be no received ACK packet to process). *Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016) (“[C]ombinations of prior art that sometimes meet the claim elements are sufficient to show obviousness.”); *Power Integrations, Inc. v. Fairchild Semiconductor Int’l, Inc.*, 843 F.3d 1315, 1336–37 (Fed. Cir. 2016) (same).

Petition for Post Grant Review
Patent No. 10,742,774

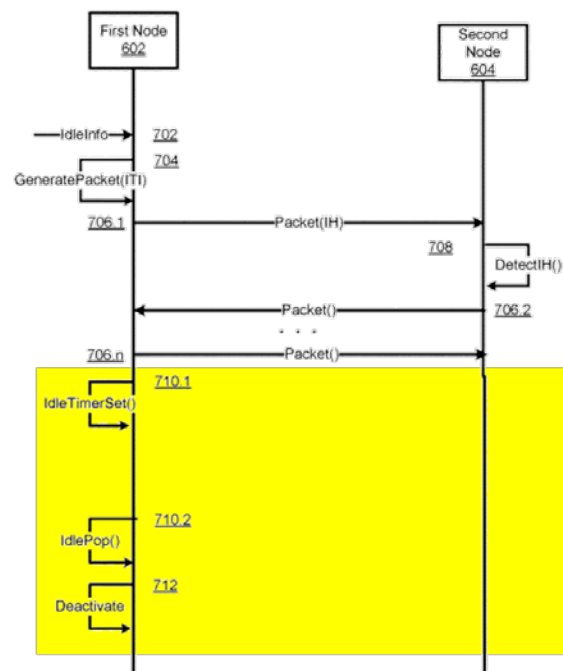


Fig. 7

(EX1001, FIG. 7 (annotated); EX1002 ¶244.)

* * *

Accordingly, for reasons similar to those explained above, the *Wookey-Eggert* combined process would have been further configured to include the process of **client machine 10** identifying an abort timeout value (“metadata”) in the abort timeout field (“time period parameter field”) for a time period, during which no packet is received from **remote machine 30’** (“first node”) in a data stream of the TCP-variant connection and processed by **client machine 10** (“second node”) to keep the TCP-variant connection active. (See *supra* Sections XI.B.1.c.2; EX1002

¶245.) Thus, the *Wookey-Eggert* combination discloses and/or suggests limitation 1.e. (EX1002 ¶246.)

- f) **calculating, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection for being used to at least partially close the TCP-variant connection.**

The *Wookey-Eggert* combination discloses and/or suggests this limitation. (EX1002 ¶¶247–256.) As discussed below, the *Wookey-Eggert* combination discloses and/or suggests calculating, by **client machine 10** (“second node”) and based on the abort timeout value (“metadata”), a timeout attribute associated with the TCP-variant connection in at least two ways. (*Id.*) Moreover, this timeout attribute is used to at least partially close the TCP-variant connection in the *Wookey-Eggert* combination. (*Id.*; see also *supra* Section XI.B.1.c–e.)

For instance, *Eggert* discloses that while “[m]any TCP implementations default to user timeout values of a few minutes” (EX1006, 3), after successful negotiation, hosts use the abort timeout for the connection. (*Id.*, 7 (“If the next reply segment does not contain an Abort Timeout Option, the connection MUST use the default abort timeout. If it does, the connection MUST use the abort timeout contained inside the Abort Timeout Option.”).) Figure 2 discloses the two scenarios for ATO negotiation, both of which disclose claim limitation 1.f.

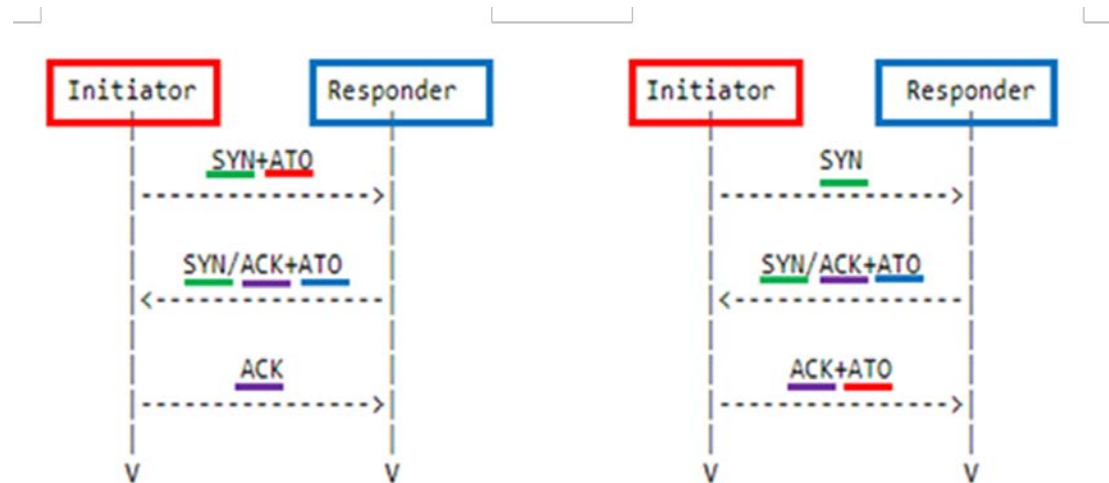


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

(*Id.*, FIG. 2 (annotated); EX1002 ¶249.)

First, *Eggert* explains that the **initiator** in the left-hand side scenario of Figure 2 has only a single option because it proposed the abort timeout value. (EX1006, 5–7.) Namely, the **initiator** “MUST be prepared to accept a shorter timeout value [from the **responder**] than proposed after the negotiation.” (*Id.*, 5.) *Eggert*’s disclosure of the **initiator** setting the user timeout value for the TCP-variant connection as the abort timeout value chosen by the **responder** describes a “calculate[ing]” process like that recited in limitation 1.g. (EX1002 ¶250.)

Second, as shown on the right-hand side of Figure 2, because the **responder** specified the initial ATO in the “SYN/ACK+ATO” segment, the **initiator** may decide “whether to accept, shorten, or reject its peer’s proposed abort timeout.” (EX1006, 5–7; *see also id.*, 9.) Thus, the **initiator** calculates the user timeout value

Petition for Post Grant Review
Patent No. 10,742,774

for the TCP-variant connection by setting it based on accepting, shortening, or rejecting the value (“metadata”) in the ATO field in the “SYN/ACK+ATO” segment. (EX1002 ¶251.)

This understanding that *Eggert*’s descriptions regarding setting the timeout attribute for the TCP-variant connection based on the negotiated abort timeout value teach calculating the timeout attribute like that claimed is consistent with PO’s view of the same claimed features. (EX1017, 10 (PO alleging that the “calculating” feature is met “since the *idle_timeout* value *sets* the duration of the idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified *based on the value received in the idle_timeout field of the connection negotiation packet*.”).) Moreover, the ’774 patent describes timeout attributes broadly and is consistent with *Eggert*’s abort timeout value used to set the user timeout. (EX1001, 22:28–35 (“ITP option handler component 562 may [*sic*] one or more attribute option handler components 564 to modify one or more corresponding attributes of a keep-alive option, a TCP user timeout, a retransmission timeout, an acknowledgment timeout, and another timeout associated with the TCP connection, in response to identifying the ITP header”); EX1002 ¶252.)

In addition, in the *Wookey-Eggert* combination, the user timeout value (“timeout attribute”) set for the TCP-variant connection is used to at least partially close the TCP-variant connection. (EX1002 ¶253.) For example, as discussed

Petition for Post Grant Review
Patent No. 10,742,774

above, the user timeout in *Eggert* is used to abort (“at least partially close”) the TCP-variant connection. (*See supra* Sections XI.B.1.c.1, XI.B.1.e.) In particular, the abort timeout value that is negotiated in *Eggert* via the ATO is used to set the user timeout value for the TCP-variant connection established between the two nodes. (*Id.*; EX1006, 1 (“The TCP Abort Timeout Option allows conforming TCP implementations to negotiate individual, per-connection abort timeouts.”).) When there is no ACK packet received with respect to a previously sent segment for the duration of the set user timeout, the TCP-variant connection will abort (“at least partially close”). (EX1006, 3 (“The TCP specification [1] includes a ‘user timeout’ that defines the maximum amount of time that segments may remain unacknowledged before TCP will abort the connection. *If a disconnection lasts longer than the user timeout, the TCP connection will abort.*”), 5 (“The timeout value included in the option specifies *the proposed abort timeout for the connection.*”).) Therefore, the user timeout value (“timeout attribute”) set for the TCP-variant connection is used to abort (“at least partially close”) the TCP-variant connection. (EX1002 ¶254.)

Accordingly, for reasons similar to those explained above, the *Wookey-Eggert* combined process would have been further configured to include the process of calculating, by **client machine 10** and based on the abort timeout value (“metadata”), a user timeout value (“timeout attribute”) associated with the TCP-

variant connection for being used to abort (“at least partially close”) the TCP-variant connection. (*See supra* Sections XI.B.1.c.2; EX1002 ¶255.) Thus, the *Wookey-Eggert* combination discloses and/or suggests limitation 1.f. (EX1002 ¶256.)

XII. EGGERT IS A PRINTED PUBLICATION

Eggert is an IETF Internet-Draft (“ID”) and is prior art under at least AIA 35 U.S.C. § 102(a)(1) because it was published in April 2004. The declaration of Alexa Morris, Managing Director of IETF, confirms *Eggert* was published, disseminated, and reasonably available to the public by April 2004. (EX1019 ¶¶1–3, 9–10.)

IDs were at the time, and continue to be, working documents published through the IETF Secretariat and disseminated to the public by IETF through various media for public comment. (*Id.* ¶¶4–6.) Since 1998 (including 2004 and today), the IETF Secretariat publishes an ID on its public website, and publication announcements were sent to an IETF mailing list and relevant working group mailing lists. (*Id.* ¶¶5–6.) Anyone could have subscribed to IETF mailing lists, and the archives of all IETF mailing lists are publicly available on IETF’s website. (*Id.* ¶¶6–8.)

Eggert was published by the IETF in April 2004. (*Id.* ¶¶9–10.) In 2004, a POSITA could have learned about *Eggert* in various ways, such as through announcements on the IETF announce mailing list, discussions on the IETF tcpm

Petition for Post Grant Review
Patent No. 10,742,774

(TCP Maintenance and Minor Extensions) working group mailing list, or review of the archives of the IETF announce or tcpm mailing lists. (*Id.*)

Collectively, this demonstrates that *Eggert* was published and publicly available in 2004 and at least prior to 2010. *Polycom, Inc., v. Directpacket Research, Inc.*, IPR2019-01235, Paper 19, 30–32 (Jan. 13, 2020) (finding a similar IETF ID was publicly available based on similar evidence).

XIII. DISCRETIONARY DENIAL IS NOT APPROPRIATE

A. 35 U.S.C. § 325(d) Does Not Provide a Basis for Discretionary Denial

Under 35 U.S.C. § 325(d), the Board considers: (1) whether the same or substantially the same art was previously presented to the Office or whether the same or substantially the same arguments previously were presented to the Office; and (2) if the first part is satisfied, whether petitioner has demonstrated that the Office erred in a manner material to the patentability of challenged claim 1. *Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6 at 8 (February 13, 2020) (precedential). Material error may include a scenario where “the Office’s previous consideration of the art is not well developed or silent.” *Id.* at 10; *see also id.* at 8–9 (n.9). In this regard, the Office has routinely instituted trial where references in an IPR were considered in an IDS but not relied upon to reject the claims during prosecution. *See, e.g., Amber.IO, Inc. D/B/A Two Tap v. 72Lux, Inc. D/B/A Shoppable*, IPR2020-00015, Paper 8 at 18–20 (April 1, 2020) (declining

Petition for Post Grant Review
Patent No. 10,742,774

to exercise discretion under § 325(d) where references were cited in an IDS but no evidence in the record that the references were substantively considered) (citing *Advanced Bionics*); *Apple, Inc. v. Omni Medsci, Inc.*, IPR2020-00029, Paper 7 at 52–55 (April 22, 2020) (similar).

Eggert was cited in an IDS along with forty-five other references during prosecution, with no explanation regarding any relevant disclosures. (EX1004, 120; *see also id.*, 114–121.) While the IDS citing *Eggert* was considered by the Office (EX1004, 97), *Eggert* was not relied upon during prosecution to reject the claims (*see generally id.*). *See, e.g., Amber.IO, Inc. D/B/A Two Tap* at 18–20; *Apple, Inc.* at 52–55; *Paragon 28, Inc. v. Wright Medical Technology, Inc.*, IPR2019-00896, Paper 16 at 30 (Oct. 14, 2019) (“Moreover, ‘[t]he Board has consistently declined exercising its discretion under Section 325(d) when the only fact a Patent Owner can point to is that a reference was disclosed to the Examiner during the prosecution,’” citing *Amgen Inc. v. Alexion Pharm., Inc.*, IPR2019-00739, Paper 15 at 58–59 (Aug. 30, 2019) (collecting cases)). Nor was *Eggert* previously considered in the light being presented herein, e.g., with the support of expert testimony in the proposed combination with *Wookey* as set forth in Ground 2 in Section XI.B. Accordingly, the Board should institute trial.

Eggert is also not cumulative to any prior art considered during prosecution of the ’774 patent and no arguments similar to those contained herein were ever

Petition for Post Grant Review
Patent No. 10,742,774

presented to or considered by the Office. Nonetheless, PO may assert here (as it did in the Google -845 IPR) that RFC 5482 (EX1016) listed on the face of the '774 patent is substantially similar to *Eggert*. It is not, as the Board found in the Google -845 IPR. Google -845 IPR, Paper 16 at 14–17. The Board should reach the same conclusion here.

First, like *Eggert*, RFC 5482 was not relied on to reject claims during prosecution. As noted above, the Board routinely institutes trial where references in an IPR were considered in an IDS but not relied upon to reject claims during prosecution. *See, e.g., Amber.IO, Inc. D/B/A Two Tap*, IPR2020-00015, Paper 8 at 18–20; *Apple, Inc.*, IPR2020-00029, Paper 7 at 52–55. Consistent with those proceedings, RFC 5482 was cited during prosecution along with forty-five other references in the same IDS citing *Eggert* with no explanation regarding any relevant disclosures. (EX1004, 120; *see also id.*, 114–121.) But RFC 5482 was never relied upon to reject the claims. (*See generally id.*) Even assuming *Eggert* is cumulative to RFC 5482, which it is not as explained below, *Eggert* was not considered in the light being presented herein as explained above (e.g., with supporting expert testimony).

Second, as the Board previously found, *Eggert*'s protocol is fundamentally different from that discussed in RFC 5482 because, in contrast with RFC 5482, *Eggert*'s protocol calls for a common negotiated value for the timeout. Google -845

Petition for Post Grant Review
Patent No. 10,742,774

IPR, Paper 16 at 14–17. While *Eggert* discloses a negotiation protocol, RFC 5482 was purposefully designed such that “an exchange of UTO [user timeout] options between both ends of a connection is not a binding negotiation.” (EX1016, 4.)

Eggert also has meaningful disclosures that do not appear in RFC 5482. For example, it is *Eggert*’s mechanisms disclosed with reference to Figure 2 regarding exchange of ATO information during the three-way handshake that is relied upon to teach claim limitations 1.c–f. (*Supra* Sections XI.B.1.c–f.) In contrast, RFC 5482 “does not define a mechanism to negotiate support of the TCP User Timeout Option [UTO] during the three-way handshake.” (EX1016, 9.) Therefore, these “material differences between the asserted art and the prior art involved during examination” weigh against discretionary denial. *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8, 17–18 (Dec. 15, 2017) (precedential as to §III.C.5, first paragraph); *see also* Google -845 IPR, Paper 16 at 16–17.

Accordingly, § 325(d) does not provide a basis for discretionary denial.

B. The Related Litigation Provides No Basis for Discretionary Denial

The Board should not exercise its discretion to deny institution under 35 U.S.C. § 324(a). *NHK Spring Co., Ltd. v. Intrix-Plex Techs, Inc.*, IPR2018-00752, Paper 8 (Sept. 12, 2018) does not apply here as no trial date has been scheduled in the related litigation. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 at 3 (Mar. 20, 2020) (precedential) (“*NHK* applies ... where the district court has set a

Petition for Post Grant Review
Patent No. 10,742,774

trial date to occur earlier than the Board’s deadline to issue a final written decision in an instituted proceeding.”). The *Fintiv* six-factor test (“*Fintiv* factor(s)”) favors institution. *See id.*, 5–6.

The **first *Fintiv* factor** (stay) is neutral, if not weighing in favor of institution. While there is currently no stay pending IPR resolution before the Western District of Texas (“WDTX”) court, Google intends to pursue such a stay should institution be granted. The Board should not speculate as to how the court would rule on the requested stay based on actions taken in different cases. *Western Digital Corp. et al. v. Martin Kuster*, IPR2020-01391, Paper 10 at 8–9 (February 16, 2020) (finding this factor neutral despite Patent Owner assertions regarding WDTX Judge Albright’s past practices regarding stays). Moreover, Google moved to transfer the WDTX litigation to the Northern District of California (“NDCA”). The WDTX litigation is stayed pending resolution of Google’s transfer motion. (EX1035.)

The **second *Fintiv* factor** (proximity of trial dates) weighs strongly in favor of institution. *First*, the district court has not set a trial date, which “weighs significantly against exercising [] discretion to deny institution of the proceeding.” *Google LLC v. Uniloc 2017 LLC*, IPR2020-00441, Paper 13 at 35 (July 17, 2020); *Nested Bean, Inc. v. Big Beings USA PTY LTD*, IPR2020-01234, Paper 15 at 14 (Jan. 25, 2021) (same). In fact, while the parties submitted proposed schedules, the court stayed the case pending resolution of Google’s transfer motion and declined to set a

Petition for Post Grant Review
Patent No. 10,742,774

case schedule.²⁶ (EX1035; EX1029.) Accordingly, determining at this time whether the Board’s anticipated FWD date (e.g., October–November 2022) will occur after trial in the related litigation would be speculative.

Even if the parties’ most recent proposed schedule was considered as a baseline for comparison (which Google submits would be improper at this time), dates in that proposed schedule would need to be delayed because they were proposed prior to the court’s recent stay order and did not take the stay into consideration. (EX1035; EX1029, 5–9.) Accordingly, the proposed dates in light of the court’s stay order demonstrates that trial will likely occur after August 2022.²⁷ (EX1029, 6, 9.) Thus, any projected trial date (*albeit* speculative) would be more

²⁶ Disposition of Google’s transfer motion has taken priority over other activities. (EX1035.) *See also In re Apple Inc.*, 979 F.3d 1332, 1337 (Fed. Cir. 2020) (“disposing of a filed transfer motion ‘should unquestionably take top priority.’”).

²⁷ Consistent with the court’s practice, the proposed order stages deadlines and trial date based on the proposed August 2021 *Markman* hearing and the “actual trial date” may materially differ from the schedule and the court would consider “reasonable amendments to the case schedule post-*Markman*.” (EX1029, 9 n.3; *id.*, 6.)

Petition for Post Grant Review
Patent No. 10,742,774

likely after the expected due date of the Board's FWD (e.g., around October–November 2022).

The Board has instituted trial even where an *actual* litigation trial date precedes the FWD deadline by two months or more. *Sand Revolution II, LLC v. Cont'l Intermodal Grp–Trucking LLC*, IPR2019-01393, Paper 24 at 8–10 (June 16, 2020) (“*Sand*”) (informative) (factor 2 favored institution despite WDTX trial preceding FWD deadline by five months); *Western Digital Corp.*, IPR2020-01391 at 9–10; *Apple Inc. v. Parus Holdings, Inc.*, IPR2020-00686, Paper 9 at 11–13, 22; *SMIC, Am. v. Innovation Foundry Techs., LLC*, IPR2020-00786, Paper 10, at 20–21 (Oct. 5, 2020); *see also Fintiv*, IPR2020-00019, Paper 11 at 5, 9.

It is also unlikely that trial in WDTX will proceed without delay. For the past twelve months, WDTX has suspended almost all trials from March 13, 2020 to at least April 30, 2021 due to the COVID-19 pandemic, creating a large backlog. (*See* EX1030; *see also* EX1031, 2 (postponing trial date “to allow the COVID-19 situation to ameliorate”); EX1034 (showing J. Albright’s docket including 792 cases filed in 2020).) Thus, as the Board has recognized, civil trials in WDTX “may possibly slip ... [due to] months of backlogged trials, including many active criminal cases that would take precedence over civil trials.” *HP Inc. v. Slingshot Printing LLC*, IPR2020-01085, Paper 12 at 7 (Jan. 14, 2021); *Sand* at 8–10.

Petition for Post Grant Review
Patent No. 10,742,774

Finally, the potential for transfer based on Google’s transfer motion further reduces the likelihood that trial will precede the Board’s FWD. *Dish Network, L.L.C. v. Broadband iTV, Inc.*, IPR2020-01267, Paper 15 at 17–18 (Jan. 21, 2021) (explaining that the second *Fintiv* factor “is a proxy for the *likelihood* that the trial court will reach a decision on validity issues before the Board reaches a [FWD]”). Indeed, transfer of the WDTX case to the proper venue (the NDCA) will undoubtedly delay any trial beyond the anticipated FWD due date. *C.f. Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 8, at 7 n.7 (Mar. 27, 2020); *see Uniloc 2017 LLC v. Google LLC*, Case No. 18-CV-00502, Dkt No. 277 (E.D. Tex. June 19, 2020) (transfer order vacating unreached deadlines).

The **third *Fintiv* factor** (investment in parallel proceedings) also weighs in favor of institution. The case is in its earliest stages, and as discussed, all deadlines other than those concerning Google’s transfer motion have been stayed. Thus, investment by the parties in invalidity and claim constructions issues will be delayed, especially if the case is transferred. And even if the transfer motion is denied, using the above-discussed proposed schedule as an exemplary baseline (which will require adjustments), any *Markman* hearing in the case is likely not to occur until August 2021 or later, and thus would occur near the expected due date of the Board’s institution decision. (EX1029, 6.) Discovery will not commence, and dispositive motions will not be due, until after the *Markman* hearing. Other case activity that

Petition for Post Grant Review
Patent No. 10,742,774

has taken place or will eventually take place (outside claim construction) that does not relate to validity should “not weigh in [the Board’s] consideration of this issue.” *Western Digital Corp.*, IPR2020-01391 at 11. Such minimal investment is insufficient to support discretionary denial. *Juniper Networks, Inc. v. Huawei Digital Techs. (Cheng Du) Co., Ltd.*, IPR2020-01130, Paper 13 at 12–13 (Jan. 22, 2021); *Dish* at 19–21; *HP* at 7. Additionally, Google’s diligence in filing this Petition just five months after receiving PO’s narrowed list of asserted claims²⁸ further weighs against discretionary denial. *Dish* at 20–21; *Fintiv* at 11.

The **fourth *Fintiv* factor** (overlap) is neutral if not in favor of institution. There is, at this early stage of the district court litigation, no evidence of overlap with the grounds here and any invalidity positions to be pursued in the district court.

²⁸ PO initially asserted 455 claims across eight patents (including 79 claims from the ’774 patent). (EX1037, 1.) The WDTX court ordered PO to substantially reduce the asserted claims to approximately 65 claims. On October 20, 2020, PO narrowed its asserted claims as ordered, but “reserve[d] the right to either narrow the claims further, **substitute claims**, or **add a reasonable number of claims**.” (EX1032.) On December 4, 2020, PO informed Petitioner that it will not modify its narrowed list of asserted claims “**at this time**.” (EX1033.)

Petition for Post Grant Review
Patent No. 10,742,774

Indeed, preliminary invalidity contentions have not yet been served. And, as discussed above, the district court must first resolve the pending transfer motion before addressing substantive issues, including invalidity. Given the stay of the litigation, it is likely expert discovery relating to invalidity will not be completed until at the earliest mid-2022 and dispositive motions on those issues would be due after that. Accordingly, substantive invalidity issues would not be addressed in the litigation until well past the institution decision and likely very near the time of the Board's expected FWD. Moreover, Petitioner may rely on invalidity grounds not asserted here in the district court litigation, including system art and prior art references and combinations not raised herein. Therefore, any potential overlap between this proceeding and the district court is minimal and speculative at this time.

Other circumstances (*Fintiv* factor 6) also favor institution. The Petition presents strong grounds demonstrating the unpatentability of the challenged claim, weighing against discretionary denial. (*See supra* Section XI.) *Western Digital Corp.*, IPR2020-01391 at 14–15. Indeed, the Board has already instituted the Google -845 IPR and the Unified -742 IPR challenging claims of the related '995 patent. (*See supra* Section II.) The Google -845 IPR is based on *Eggert*, which is being applied here. Moreover, this petition is the only challenge to the '774 patent before the Board, which is a “crucial fact” favoring institution. *Google LLC v. Uniloc 2017 LLC*, IPR2020-00115, Paper 10 at 6 (May 12, 2020).

Petition for Post Grant Review
Patent No. 10,742,774

While **Fintiv factor 5** (same parties) may weigh slightly in favor of denial, the remaining factors are at least neutral, if not strongly favoring institution. Further, even if the Board finds that **factor 1** (or even **factor 2**) somehow tilts in favor of denial, Petitioner’s diligence, the lack of relevant investment of resources, lack of evidence of overlap, and/or strength of Petitioner’s grounds (**factors 3, 4, 6**) outweigh these other factors. *See e.g., SK Hynix Inc. et al. v. Netlist, Inc.*, IPR2020-01421, Paper 10 at 6–13 (Mar. 16, 2021). Accordingly, based on a “holistic view” of whether integrity of the system and efficiency is best served, institution here is proper. *Samsung Elecs. Co. Ltd. v. Dynamics Inc.*, IPR2020-00505, Paper 11 at 15 (Aug. 12, 2020).

Petition for Post Grant Review
Patent No. 10,742,774

XIV. CONCLUSION

For the reasons given above, Petitioner requests institution of PGR for claim 1 of the '774 patent, and a finding that claim 1 is unpatentable based on the above grounds.

Respectfully submitted,

Dated: May 10, 2021

By: Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned certifies that the foregoing Petition for Post Grant Review of U.S. Patent No. 10,742,774 contains, as measured by the word processing system used to prepare this paper, 18,679 words. This word count does not include the items excluded by 37 C.F.R. § 42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: May 10, 2021

By: Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

CERTIFICATE OF SERVICE

I hereby certify that on May 10, 2021, I caused a true and correct copy of the foregoing Petition for Post Grant Review of U.S. Patent No. 10,742,774 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on PAIR:

The Caldwell Firm, LLC
Patrick Caldwell (Reg. No. 44580)
G. Gordon (Reg. No. 64517)
P.O. Box 59655
Dept. SVIPGP
Dallas, TX 75229

A courtesy copy was also sent electronically to Patent Owner's litigation counsel listed below:

TD-PTAB@devlinlawfirm.com
ddahlgren@devlinlawfirm.com
dlflitparas@devlinlawfirm.com
1526 Gilpin Ave.
Wilmington, DE 19806

Respectfully submitted,

Dated: May 10, 2021

By: /Naveen Modi/
Naveen Modi (Reg. No. 46,224)
Counsel for Petitioner

EXHIBIT 12

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC, LG ELECTRONICS, INC., and
LG ELECTRONICS U.S.A., INC.,
Petitioner,

v.

JENAM TECH, LLC,
Patent Owner.

IPR2020-00845
Patent 9,923,995 B1

Before DANIEL J. GALLIGAN, SCOTT B. HOWARD, and
JASON M. REPKO, *Administrative Patent Judges*.

GALLIGAN, *Administrative Patent Judge*.

JUDGMENT
Final Written Decision
Determining All Challenged Claims Unpatentable
35 U.S.C. § 318(a)

IPR2020-00845

Patent 9,923,995 B1

I. INTRODUCTION

In this *inter partes* review, Google LLC, LG Electronics, Inc., and LG Electronics U.S.A., Inc. (“Petitioner”) challenge the patentability of claims 1–24 of U.S. Patent No. 9,923,995 B1 (“the ’995 patent,” Ex. 1001), which is assigned to Jenam Tech, LLC (“Patent Owner”).

We have jurisdiction under 35 U.S.C. § 6. This Final Written Decision, issued pursuant to 35 U.S.C. § 318(a), addresses issues and arguments raised during the trial in this *inter partes* review. For the reasons discussed below, we determine that Petitioner has proven by a preponderance of the evidence that claims 1–24 of the ’995 patent are unpatentable. *See* 35 U.S.C. § 316(e) (2018) (“In an *inter partes* review instituted under this chapter, the petitioner shall have the burden of proving a proposition of unpatentability by a preponderance of the evidence.”).

A. Procedural History

Petitioner requested *inter partes* review of claims 1–24 of the ’995 patent on the following grounds:

Claim(s) Challenged	35 U.S.C. § ¹	Reference(s)/Basis
1–24	103	Eggert ²
1–24	103	Eggert, Hankinson ³

¹ The Leahy-Smith America Invents Act (“AIA”) included revisions to 35 U.S.C. §§ 102 and 103 that became effective March 16, 2013. The application for the ’995 patent was filed on September 3, 2017. Although the ’995 patent claims priority to applications filed before March 16, 2013, Patent Owner has not shown that the written description of the earlier applications supports the challenged claims. *See Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378–80 (Fed. Cir. 2015); *Tech. Licensing Corp. v. Videotek, Inc.*, 545 F.3d 1316, 1327 (Fed. Cir. 2008). Therefore, we apply the post-AIA versions of 35 U.S.C. §§ 102 and 103.

² L. Eggert, “TCP Abort Timeout Option,” Apr. 14, 2004 (Ex. 1004).

³ Hankinson, EP 1 242 882 B1, published Apr. 20, 2005 (Ex. 1005).

IPR2020-00845

Patent 9,923,995 B1

Claim(s) Challenged	35 U.S.C. § ¹	Reference(s)/Basis
16	103	Eggert, Hankinson, RFC 1122 ⁴

Paper 3 (“Pet.”), 11. Patent Owner filed a Preliminary Response. Paper 8. With our authorization, Petitioner filed a Reply to the Preliminary Response (Paper 13) addressing 35 U.S.C. § 325(d). We instituted trial on all grounds of unpatentability. Paper 16 (“Inst. Dec.”), 27.

During the trial, Patent Owner filed a Response (Paper 18, “PO Resp.”), Petitioner filed a Reply (Paper 23, “Pet. Reply”), and Patent Owner filed a Sur-reply (Paper 27, “PO Sur-reply”).

An oral hearing was held on July 8, 2021, a transcript of which appears in the record. Paper 32 (“Tr.”).

B. Related Matters

As required by 37 C.F.R. § 42.8(b)(2), the parties identify various related matters. Pet. 77; Paper 5, 1. A different petitioner (Unified Patents, LLC) challenged claims of the ’995 patent in IPR2020-00742, but that case was terminated due to settlement after institution of trial. *Unified Patents LLC v. Jenam Tech LLC*, IPR2020-00742, Paper 31 (PTAB June 29, 2021).

C. Real Parties in Interest

Petitioner identifies the following real parties in interest: LG Electronics, Inc., LG Electronics U.S.A., Inc., Google LLC, Samsung Electronics Co., Ltd., and Samsung Electronics America, Inc. Pet. 76. Patent Owner identifies itself as the real party in interest. Paper 5, 1.

⁴ RFC 1122, “Requirements for Internet Hosts - - Communication Layers,” Oct. 1989 (Ex. 1007).

IPR2020-00845

Patent 9,923,995 B1

D. The '995 Patent and Illustrative Claim

The '995 patent states that some transmission control protocol (TCP) implementations support a keep-alive option, but the '995 patent states that “[e]ach node supports or does not support keep-alive for a TCP connection based on each node’s requirements without consideration for the other node in the TCP connection.” Ex. 1001, 1:43–45, 1:53–56. According to the '995 patent, “[t]o date no mechanism to allow two TCP connection endpoints to cooperate in supporting the keep-alive option has been proposed or implemented.” Ex. 1001, 2:14–17. Thus, the '995 patent states that “there exists a need for methods, systems, and computer program products for sharing information for detecting an idle TCP connection.” Ex. 1001, 2:21–23.

Claims 1 and 11 are illustrative and are reproduced below.

1. An apparatus comprising:
 - a non-transitory memory storing instructions; and
 - one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for:
 - receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established;
 - detecting an idle time period parameter field in the TCP-variant packet;
 - identifying metadata in the idle time period parameter field for an idle time period and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active; and
 - modifying, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection.

IPR2020-00845

Patent 9,923,995 B1

11. The apparatus of claim 1 wherein the one or more processors execute the instructions for:

detecting the idle time period based on the timeout attribute; and

in response to detecting the idle time period, deactivating the TCP-variant connection by releasing a resource allocated for the TCP-variant connection by one of the first and second nodes without signaling another one of the first and second nodes that is related to the detection of the idle time period.

II. ANALYSIS

A. Level of Ordinary Skill in the Art

Petitioner contends that a person of ordinary skill in the art

would have an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking, or in the alternative, equivalent experiences, such as 6 years of work or research experience in the field of networking.

Pet. 10 (citing Ex. 1003 ¶¶ 29–32). Patent Owner largely agrees with Petitioner’s assessment:

Patent Owner does not contest Petitioners’ assertion that a person of ordinary skill in the art (“POSITA”) would have had (i) an undergraduate degree in electrical engineering, computer engineering, computer science or a related field and (ii) at least two years of work experience in the field of networking. A higher level of education (e.g., a master’s degree) may make up for less work experience, and additional work experience (e.g., 5-6 years) may make up for less education.

PO Resp. 9–10 (citing Ex. 1003, 7–8⁵).

⁵ Patent Owner cites pages 7–8 of the declaration of Mr. Bradner, who is Petitioner’s declarant. Paragraphs 29–32, which Petitioner cites as support for its proposal, appear at pages 7–8.

Thus, the parties are in general agreement as to the level of ordinary skill in the art, and both cite the same testimony of Petitioner’s declarant in support. We note that Petitioner’s alternative qualifications specify the field of networking, whereas Patent Owner’s do not, although we view this as implied by Patent Owner’s argument. Because Petitioner more precisely specifies the level of skill and because Petitioner’s proposed level of skill is supported by evidence cited by both parties, we adopt Petitioner’s description, with the exception of the open-ended language “at least.” *See* Ex. 1003 ¶¶ 30–32.

B. Claim Construction

We interpret claim terms using “the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. § 42.100(b) (2019).

1. Idle Time Period

The focus of this trial is whether or not the asserted prior art teaches an “idle time period,” as recited in claims 1 and 15. Claim 1 recites “identifying metadata in the idle time period parameter field for an *idle time period and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active*,” and claim 15 recites “receiving idle information for detecting an *idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active*” (emphases added).

Patent Owner provides the following table setting forth its proposed constructions for the claim recitations italicized above:

IPR2020-00845

Patent 9,923,995 B1

Claim Language	Patent Owner's Proposed Construction
“idle time period and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active” (Claim 1)	“idle time period and, during which, no packet <u><i>of any kind</i></u> is <u><i>sent or received by the apparatus</i></u> in the TCP-variant connection to keep the TCP-variant connection active <u><i>with or without a condition and via any mechanism</i></u> ”
“an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active” (Claim 15)	“an idle time period, during which, no packet <u><i>of any kind</i></u> is <u><i>sent or received by the apparatus</i></u> in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active <u><i>with or without a condition and via any mechanism</i></u> ”

PO Resp. 38; *see also* PO Resp. 10–38 (argument concerning these constructions).

As set forth in the table above, Patent Owner's proposed constructions are based on the following three terms or phrases in the recited subject matter: “no packet,” “communicated,” and “to keep the TCP-variant connection active.” PO Resp. 10–38. First, as to the language “no packet,” Patent Owner argues that “[t]here are no qualifiers, adjectives, etc. stated in connection with the term ‘packet’ and none should be inserted through erroneous construction or interpretation.” PO Resp. 11–12 (citing Ex. 2020 ¶¶ 33–45). We disagree. Although the phrase “no packet” is absolute, the remainder of the claim does include a significant qualifier by reciting that no packet is communicated “to keep the TCP-variant connection active.” Thus, we have no issue with interpreting the words “no packet” in isolation to mean “no packet of any kind,” but we cannot ignore the phrase “to keep the

IPR2020-00845

Patent 9,923,995 B1

TCP-variant connection active,” which is at the center of the parties’ dispute, as discussed below. Thus, the phrase “no packet of any kind” is still qualified by other language in the claim.

Second, Patent Owner argues that the term “communicated” means “sent or received by the apparatus.” PO Resp. 15–20. Patent Owner proposes this construction “to clarify that the claims, when taking into consideration the effect of the negative ‘no packet’ limitation, properly *exclude* the possible meanings of ‘communicated’ including, but not limited to ‘only receiving,’ ‘only sending,’ and ‘*receiving and sending*.’”

PO Resp. 19–20. According to Patent Owner, “under no circumstance[] should the claims be misinterpreted to require that any packet ‘sent’ by the apparatus *must also* be ‘received’ by another node (i.e. *not* ‘only sen[t]’ per claim 30).” PO Resp. 20. Petitioner argues that Patent Owner’s proposed construction is improper but argues that the asserted prior art teaches the subject matter under Patent Owner’s interpretation in any event. Pet. Reply 2–7. As explained below, we agree with Petitioner that the asserted prior art teaches “communicated” under Patent Owner’s proposed construction. Therefore, we apply Patent Owner’s construction of “communicated,” and we need not resolve any disputes between the parties over the scope of this term. *See, e.g., Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (“[W]e need only construe terms ‘that are in controversy, and only to the extent necessary to resolve the controversy’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

Third, Patent Owner argues that the phrase “to keep the TCP-variant connection active” means “to keep the TCP-variant connection active with

IPR2020-00845

Patent 9,923,995 B1

or without a condition and via any mechanism.” PO Resp. 20–38. Patent Owner explains that this means keeping the connection active

regardless as to what condition, timer, period, counter, or other mechanism exists to keep the connection active. Put differently, if *any* packet is sent or received by the apparatus for the purpose of keeping the connection active regardless of how it is designed to keep the connection active, this is outside the scope of the claimed idle time period.

PO Resp. 21 (citing Ex. 2020 ¶¶ 54–65). Thus, Patent Owner’s construction asks us to look at the purpose for which a packet is communicated. As explained below, we find that the asserted prior art teaches, in at least one scenario, an absence of packets that are communicated for the purpose of keeping the connection active, and, therefore, we apply Patent Owner’s construction of the phrase “to keep the TCP-variant connection active” and need not resolve any disputes regarding the scope of this phrase.

2. *Remaining Terms*

For purposes of this Decision, we need not construe expressly any other claim terms. *See Nidec*, 868 F.3d at 1017.

C. *Principles of Law*

A patent claim is unpatentable under 35 U.S.C. § 103 if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) any secondary

IPR2020-00845

Patent 9,923,995 B1

considerations, if in evidence.⁶ *Graham v. John Deere Co. of Kan. City*, 383 U.S. 1, 17–18 (1966).

*D. Alleged Obviousness over Eggert
(Claims 1–24)*

Petitioner asserts that claims 1–24 of the ’995 patent are unpatentable under 35 U.S.C. § 103 as obvious over the teachings of Eggert. Pet. 18–61. Patent Owner opposes. PO Resp. 41–55. For the reasons explained below, we determine that Petitioner has proven by a preponderance of the evidence that claims 1–24 are unpatentable as obvious over Eggert.

1. Eggert

Eggert “specifies a new TCP option - the Abort Timeout Option - that allows conforming hosts to negotiate per-connection abort timeouts. This allows mobile hosts to maintain TCP connections across disconnected periods that are longer than their system’s default abort timeout.”

Ex. 1004, 3. Petitioner argues that Eggert was published on April 14, 2004, which is more than one year before the earliest priority date of the ’995 patent, and therefore qualifies as prior art under 35 U.S.C. § 102. Pet. 12–14 (citing Ex. 1003 ¶¶ 53–54, 67, 73–80, 114–120–126, 128–138; Ex. 1028; Ex. 1029; Ex. 1031, 4–10). Patent Owner does not challenge the prior art status of Eggert. *See generally* PO Resp. We are persuaded that Petitioner has shown that Eggert qualifies as prior art under 35 U.S.C. § 102. *See* Pet. 12–14.

2. Independent Claim 1

Independent claim 1 is directed to “[a]n apparatus comprising: a non-transitory memory storing instructions; and one or more processors in

⁶ Patent Owner does not present any objective evidence of nonobviousness (i.e., secondary considerations) as to any of the challenged claims.

IPR2020-00845

Patent 9,923,995 B1

communication with the non-transitory memory, wherein the one or more processors execute the instructions for” performing four operations listed in claim 1. Petitioner argues that Eggert’s reference to Solaris would have been understood by a person of ordinary skill in the art as describing a Solaris computer with the Solaris Operating System and a TCP implementation. Pet. 18 (citing Ex. 1004, 3⁷, § 2.1; Ex. 1003 ¶ 163). Petitioner further contends that it would have been obvious to a person of ordinary skill in the art that a hard disk (the recited “non-transitory memory”) in the Solaris computer stores instructions and that a CPU (the recited “one or more processors”) would execute instructions that are stored in the hard disk. Pet. 18–20 (citing Ex. 1004, 3, 5–8 (§ 2.1); Ex. 1003 ¶¶ 163–189).

Patent Owner does not dispute Petitioner’s contentions for the claimed subject matter quoted above, but Patent Owner argues that Eggert does not teach the “idle time period” subject matter recited in one of the four operations. *See* PO Resp. 41–54. We address Patent Owner’s “idle time period” arguments below.

We are persuaded by Petitioner’s contentions and evidence, summarized above, and we conclude that Eggert’s disclosure of a Solaris computer implementation renders obvious “[a]n apparatus comprising: a non-transitory memory storing instructions; and one or more processors in

⁷ The Petition cites the page numbers that appear as part of Eggert itself, which are in the upper right corner of every other page. Pet. 14 n.4 (“Petitioners cite to the page numbers indicated in Eggert.”). We cite the exhibit page numbers in the lower right-hand corner of Exhibit 1004.

IPR2020-00845

Patent 9,923,995 B1

communication with the non-transitory memory, wherein the one or more processors execute the instructions.”⁸

a) Receiving a TCP-variant packet

The first operation recited in claim 1 is “receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established” (referred to herein as “the receiving operation”). Petitioner argues that Eggert discloses that a responder (“second node”) receives a synchronization (SYN) packet that contains an abort timeout option (ATO) during a three-way handshake (i.e., before a connection is established). Pet. 20 (citing Ex. 1004, 5; Ex. 1003 ¶¶ 190–223). Petitioner argues that “a modified TCP SYN segment (SYN+ATO) that contains an ATO field is a TCP-variant packet, since it uses the same format as set forth in the TCP standard but adds the ATO field.” Pet. 24 (citing Ex. 1003 ¶¶ 196, 451) (footnote omitted); *see also* Pet. 22 (arguing that Eggert’s disclosure that the initiator can receive a packet containing the ATO from the responder also teaches “receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet” before a connection is established). Petitioner also argues that “[c]onnections established by exchanging ATOs are TCP-variant connections because they are established in accordance with a TCP-variant protocol by exchanging TCP-variant packets (SYN+ATO, SYN/ACK+ATO) during the three-way handshake.” Pet. 26 (citing Ex. 1003 ¶¶ 217–223).⁹

⁸ Because we find that Petitioner has demonstrated obviousness of the subject matter recited in the preambles of the challenged claims, we need not decide whether the preambles are limiting.

⁹ ACK refers to acknowledgment. *See* PO Resp. 12; Ex. 1003 ¶ 58.

During the trial, Patent Owner did not dispute Petitioner’s contentions for this subject matter. *See generally* PO Resp. 41–54 (addressing “idle time period” subject matter); *see also* Paper 17, 8 (“Patent Owner is cautioned that any arguments not raised in the response may be deemed waived.”); *In re NuVasive, Inc.*, 842 F.3d 1376, 1380–81 (Fed. Cir. 2016) (explaining that the patent owner waived an issue presented in its preliminary response that it failed to renew in its response during trial); Consolidated Trial Practice Guide 52 (Nov. 2019) (“Once a trial is instituted, the Board may decline to consider arguments set forth in a preliminary response unless they are raised in the patent owner response.”).

We are persuaded by Petitioner’s showing for the receiving operation. We find that Eggert teaches TCP-variant connections and packets because it discloses a feature to add to TCP—the abort timeout option. Ex. 1004, 3 (“This document specifies a new TCP option - the Abort Timeout Option - that allows conforming hosts to negotiate per-connection abort timeouts.”). Eggert states that “[a] TCP implementation that does not support the TCP Abort Timeout Option SHOULD silently ignore it.” Ex. 1004, 7, *quoted in* Pet. 23. “Thus,” Petitioner correctly asserts, “Eggert discloses adding a new TCP option to the traditional TCP implementation.” *See* Pet. 23 (citing Ex. 1003 ¶¶ 193–195, 214–223).

We also find that Eggert teaches receiving a TCP-variant packet “in advance of a TCP-variant connection being established,” as recited in claim 1, because the packets used to negotiate the ATO are sent as part of the three-way handshake to establish the connection between nodes. For example, Eggert states the following: “This specification allows both the initiator of a TCP connection (i.e., the node sending the SYN) as well as the responder of a TCP connection (i.e., the node receiving the SYN) to initiate

an abort timeout negotiation during the connection's three-way handshake.”
Ex. 1004, 5, *quoted in* Pet. 20.

Therefore, based on Petitioner's persuasive contentions and evidence, discussed above, we are persuaded that Petitioner has shown that Eggert teaches “receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established.”

b) Detecting an idle time period parameter and identifying metadata

The second and third operations recited in claim 1 are “detecting an idle time period parameter field in the TCP-variant packet” and “identifying metadata in the idle time period parameter field for an idle time period and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active.” Petitioner argues that Eggert discloses detecting the abort timeout field and identifying metadata in that field (the value of the timeout) in the three-way handshake of Figure 2. Pet. 26–30 (discussing exchange of ATO values in Eggert's Figure 2). Figure 2 of Eggert is reproduced below.

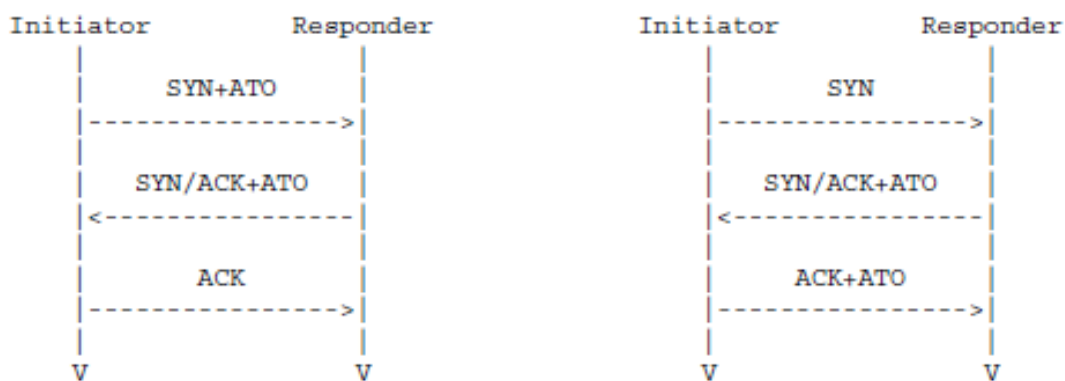


Figure 2: Allowed TCP Abort Timeout Option (ATO) Exchanges

Ex. 1004, 7. Figure 2 of Eggert, reproduced above, “illustrates the two allowed exchanges” in an abort timeout negotiation. Ex. 1004, 5.

IPR2020-00845

Patent 9,923,995 B1

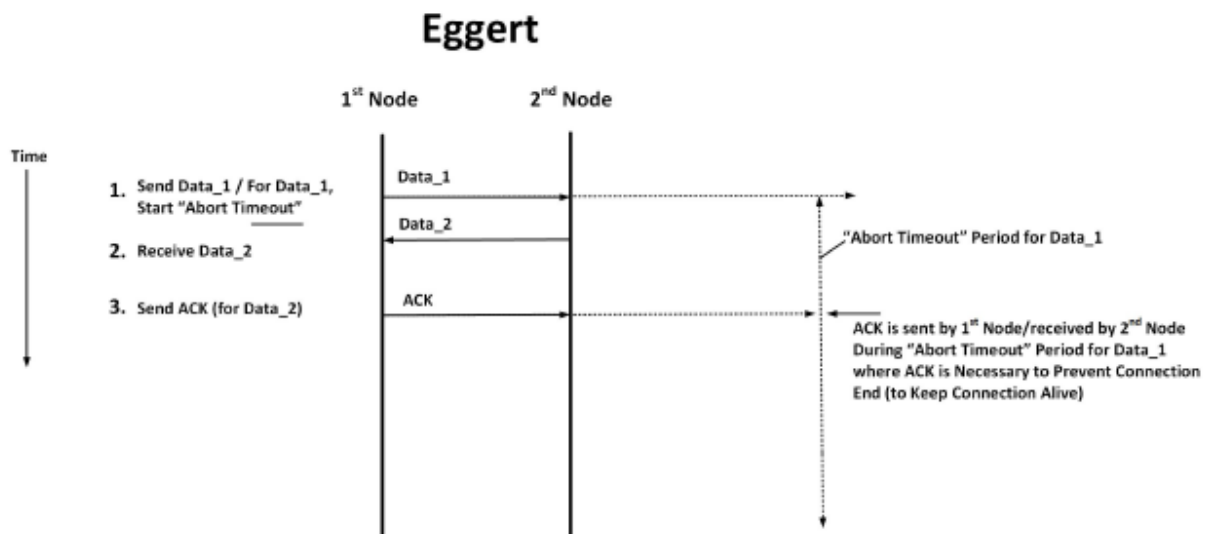
Petitioner argues that the abort timeout value contained in the ATO field that is exchanged during Eggert's three-way handshake teaches "an idle time period parameter field," as recited in claim 1. Pet. 26–29 (citing Ex. 1004, 3, 5, Fig. 2; Ex. 1003 ¶¶ 224–232). Petitioner argues that "Eggert's description of the abort timeout as the duration of time during which a node can wait to receive an ACK is a period 'during which no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active.'" Pet. 30–31 (citing Ex. 1003 ¶¶ 247–256). Petitioner argues that, in Eggert, the ACK is the packet that is communicated to keep the connection active because Eggert discloses that "the TCP specification includes a 'user timeout' that defines the maximum amount of time that segments may remain unacknowledged before TCP will abort the connection." Pet. 30 (quoting Ex. 1004, 3). Thus, according to Petitioner, when no ACK is received within a certain period of time after sending a packet, the connection will end. *See* Pet. 30.

Petitioner also argues that Eggert's disclosure that the ATO "allows mobile hosts to maintain TCP connections across disconnected periods that are longer than their system's default abort timeout" (Ex. 1004, 3) "parallels the '995 Patent, which indicates that the idle time period may be associated with physical disconnection of a network medium or simply a dead connection." Pet. 31 (citing Ex. 1004, 3, 9; Ex. 1001, 2:17–20, 11:53–62, 21:11–18; Ex. 1003 ¶¶ 247–248); *see also* Ex. 1004, 9 ("Long abort timeout values allow hosts to tolerate extended periods of disconnection.")). According to Petitioner, "no packets are communicated to keep the connection active because they cannot get through the interruption or the disconnection in the physical medium." Pet. 31 (citing Ex. 1003 ¶¶ 247–248; Ex. 1001, 2:17–20, 11:53–62, 21:11–18).

IPR2020-00845

Patent 9,923,995 B1

The parties' dispute centers on whether Eggert teaches a period during which "no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active," which we refer to as the "idle time period" or "ITP" subject matter. Patent Owner argues that Eggert does not teach the ITP subject matter because there are packets communicated to keep the connection active during the period identified by Petitioner as the idle time period. PO Resp. 41–54. Patent Owner provides the data flow diagram reproduced below to illustrate this point.



PO Resp. 45; *see also* Ex. 2020 ¶¶ 80–85 (testimony of Patent Owner's declarant, Dr. Michael Smith, regarding this data flow diagram). The diagram above illustrates a data flow sequence between two nodes in which data are sent from each of nodes 1 and 2 and an ACK is sent from node 2 to node 1. PO Resp. 45. With reference to this diagram, Patent Owner argues the following:

As shown, at step 1., the 1st Node sends Data_1, and starts the "Abort Timeout" period (or allows an existing abort timeout period to continue for Data_1). (Ex. 2020 at 78-82.) At step 2., the 1st Node receives Data_2 sent from the 2nd Node as illustrated, such that, at step 3., the 1st Node sends an ACK for

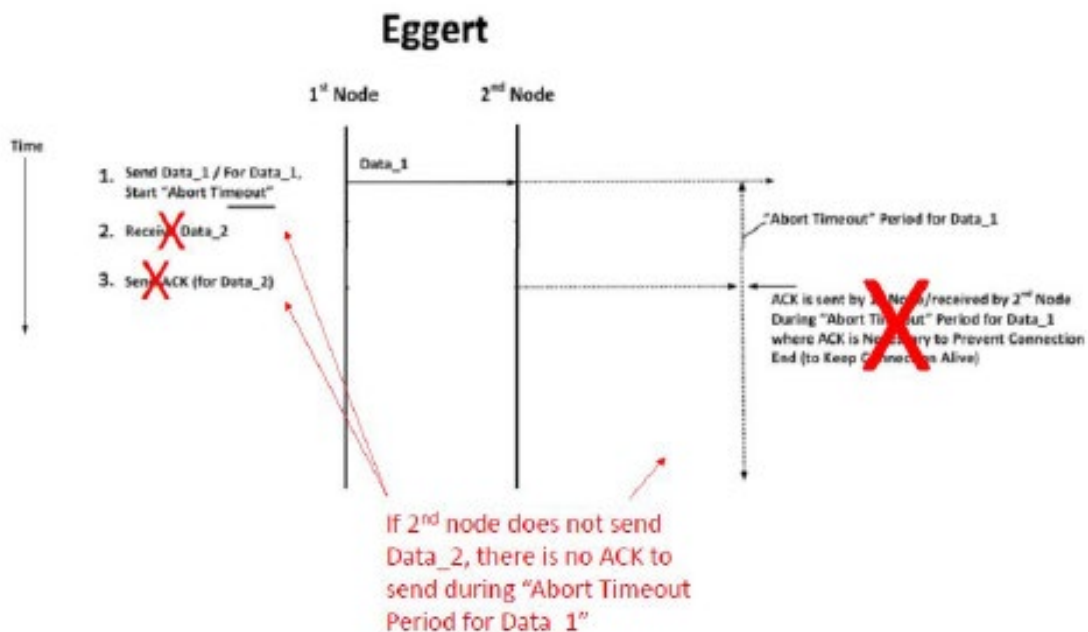
IPR2020-00845

Patent 9,923,995 B1

Data_2 *during* the “Abort Timeout” period of the 1st Node for Data_1 (i.e. before the expiration thereof). (Ex. 2020 at 78-82.) The “Abort Timeout”- period of the 1st Node for Data_1 is thus detected with *at least one packet (e.g. the sent ACK at step 3.)* being communicated in the connection to keep the connection active (e.g. to avoid an “ended” connection). (Ex. 2020 at 78-82.)

PO Resp. 44–45.

Petitioner responds by arguing that Eggert does not require Patent Owner’s proposed data flow and produces the following data flow diagram to illustrate its point.



Pet. Reply 16–18. The diagram above is an altered version of Patent Owner’s proposed data flow (PO Resp. 45) in which Petitioner removes steps 2 and 3. Pet. Reply 17. In annotations in the diagram above, Petitioner notes that if the second node does not send data at step 2, then the first node does not send an acknowledgement at step 3. Pet. Reply 17. Thus, according to Petitioner, “a timeout will occur if no data packet is communicated during the idle time period that requires acknowledgement,

IPR2020-00845

Patent 9,923,995 B1

or if such a data packet is so communicated, no ACK packet is communicated during that idle time period.” Pet. Reply 16.

Patent Owner responds that Petitioner is focusing on ACKs as “the only packets capable of keeping Eggert’s connection active, which is *not* the case.” PO Sur-reply 20. In particular, Patent Owner argues that, if node 1 sends data and does not receive an ACK, then node 1 will retransmit the data on the expiration of a retransmission timer. PO Sur-reply 20–25. According to Patent Owner, “[l]ike ACKs, the retransmitted-packets are to keep the connection active by eliciting a received ACK that was missing for an initial transmission.” PO Sur-reply 25. Patent Owner contends that, “since the retransmitted-packets elicit a received ACK (that Petitioners’ admit maintains the connection), the ACK-eliciting retransmitted-packets are *also* to keep the connection active. The connection will prematurely close without them.” PO Sur-reply 25. Thus, during the oral argument, Patent Owner’s counsel argued that “[t]his really can be resolved by answering a very simple question. Are the retransmission packets communicated in Eggert to keep the connection active?” Tr. 27:16–19.

For the reasons explained below, we find that retransmission packets that may be communicated in Eggert are not packets “to keep the TCP-variant connection active” and, therefore, that Eggert teaches that “no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active,” as recited in claim 1.

There is no dispute that in Eggert, when data are first sent, the ATO timer starts, and when the ATO timer expires, the connection is terminated. *See* PO Resp. 39 (“*Eggert*’s ATO utilizes the TCP specification in which there is a ‘user timeout’ that defines the amount of time that packets may remain unacknowledged before the connection is terminated. (Ex. 1004

IPR2020-00845

Patent 9,923,995 B1

at 3.) The failure to *receive* an ACK packet triggers termination of the connection.”); *see also* Ex. 2020 ¶ 30 (stating the same). The retransmission timer of RFC 793 cited by Patent Owner is a separate timer that governs retransmission of data if an ACK is not received for the data within a particular timeout interval. *See* PO Sur-reply 22 (citing Ex. 1006, 8, 14). RFC 793 explains that, “[w]hen the TCP transmits a segment containing data, it puts a copy on a retransmission queue and starts a timer; when the acknowledgment for that data is received, the segment is deleted from the queue.” Ex. 1006, 14. However, “[i]f the acknowledgment is not received before the timer runs out, the segment is retransmitted.” Ex. 1006, 14. Eggert does not disclose, and Patent Owner does not argue, that retransmitting data resets the original timeout timer that started upon sending the data the first time. Thus, despite retransmitting the data, the connection will terminate on the expiration of the ATO timer that started when the first data were transmitted.

As discussed above in § II.B.1, Patent Owner argues that the language “to keep the TCP-variant connection active” describes “a purpose” of the packet. *See* PO Resp. 25. If the purpose of retransmitting the data were to keep the connection active, then the device would be programmed to reset the ATO timer upon retransmission to avoid termination of the connection. The fact that this is not done supports a finding that the purpose of these packets is not to keep the connection active. Indeed, during oral argument, Patent Owner’s counsel stated that “the programming [of a computer] is done in a specific way to try to achieve a specific result.” Tr. 46:16–18. Thus, if the purpose of particular programming were to keep the connection active, the device would be so programmed by resetting the ATO timer.

Furthermore, Patent Owner’s position is that, “since the retransmitted-

For all of these reasons, we find that retransmission packets in Eggert are not communicated “to keep the TCP-variant connection active.” In view of this finding, we turn back to the following data flow posited by Petitioner:



IPR2020-00845

Patent 9,923,995 B1

Pet. Reply 17. The diagram above is an altered version of Patent Owner’s proposed data flow (PO Resp. 45) in which Petitioner removes steps 2 and 3.

Pet. Reply 17. In annotations in the diagram above, Petitioner notes that if the second node does not send data at step 2, then the first node does not send an acknowledgement at step 3. Pet. Reply 17.

In the above scenario, the first node sends “Data_1” at step one and no ACK is received. According to Dr. Smith, there can be situations “when a sent packet is not received (e.g. due to a network condition)” or “when no ACK is sent for a received data packet (e.g. due to an error condition in the receiving node).” Ex. 2020 ¶¶ 106, 108, 110, 112, 114, 116, 119, 122, 125. As discussed above, Eggert expressly contemplates network disconnections. Ex. 1004, 3. If the second node does not receive “Data_1” because of a disconnection, then the second node would not attempt to send an ACK as it would only acknowledge received packets. Even if “Data_1” were received at the second node, the second node may not send an ACK “due to an error condition in the receiving node,” as stated by Dr. Smith. Ex. 2020 ¶¶ 106, 108, 110, 112, 114, 116, 119, 122, 125. In these cases, no ACK is sent or received, consistent with Patent Owner’s proposed construction of “communicated.” See PO Resp. 15–20. The ATO timer continues to run, and in the scenario above, neither node sends additional data. The only other transmission, then, that might occur before the ATO timer expires and the connection is terminated is a retransmission of “Data_1.”¹⁰ But as

¹⁰ Depending on the negotiated ATO value, a retransmission might not even happen because Eggert “does not restrict the range of timeout values used with the TCP Abort Timeout Option. The 32-bit value in the option can express abort timeouts from zero seconds to over 136 years.” Ex. 1004, 9. Eggert then explains that “[v]ery short timeout values can affect TCP retransmissions over high-delay paths” and that, “[a]lthough the TCP Abort

IPR2020-00845

Patent 9,923,995 B1

discussed above, this retransmission is not “to keep the TCP-variant connection active,” as recited in claim 1. If an ACK is not received by the first node within the ATO period, the connection will terminate. Therefore, Eggert teaches, in the above scenario, “an idle time period and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active.”

Based on the foregoing, we disagree with Patent Owner’s assertion that “the ITP limitation is not disclosed in any scenario involving any of the references.” *See* PO Sur-reply 26. Rather, for the reasons given above, we are persuaded by Petitioner’s contention that, under at least certain operating conditions, Eggert teaches the recited ITP subject matter, which is sufficient to show obviousness. *See Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016) (“[C]ombinations of prior art that sometimes meet the claim elements are sufficient to show obviousness.”); *Hewlett–Packard Co. v. Mustek Sys., Inc.*, 340 F.3d 1314, 1326 (Fed. Cir. 2003) (“[A] prior art product that sometimes, but not always, embodies a claimed method nonetheless teaches that aspect of the invention.”).

We also are persuaded by Petitioner’s contention that Eggert teaches detecting the abort timeout field and identifying metadata in that field (the value of the timeout) in the three-way handshake of Figure 2. *See* Pet. 26–30 (discussing exchange of ATO values in Eggert’s Figure 2). For example, in discussing the negotiation of the abort timeout option, Eggert states that

Timeout Option allows negotiation of shorter timeouts, applications requesting such short timeouts should consider these effects.” Ex. 1004, 9. Thus, Eggert’s ATO timer may be programmed to be shorter than the retransmission timer, which Eggert specifically mentions is a factor that should be considered in selecting short timeout values.

IPR2020-00845

Patent 9,923,995 B1

“[t]he timeout value included in the option specifies the proposed abort timeout for the connection.” Ex. 1004, 5, *quoted in* Pet. 26. Eggert further discloses that, “[u]pon receipt of a segment with the Abort Timeout Option, the receiving host decides whether to accept, shorten, or reject its peer’s proposed abort timeout.” Ex. 1004, 5, *quoted in* Pet. 26. Thus, we find that Eggert teaches the “detecting” and “identifying” operations recited in claim 1.

For the reasons discussed above and given in the Petition, we are persuaded that Petitioner has shown, by a preponderance of the evidence, that Eggert teaches “detecting an idle time period parameter field in the TCP-variant packet” and “identifying metadata in the idle time period parameter field for an idle time period and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active.”

c) Modifying a timeout attribute

The fourth operation recited in claim 1 is “modifying, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection.” Petitioner argues that Eggert’s disclosure of shortening the proposed abort timeout value during the three-way handshake negotiation teaches this subject matter. Pet. 32–35 (citing Ex. 1004, 5–7 (§ 2.1); Ex. 1003 ¶¶ 258–269).

During the trial, Patent Owner did not dispute Petitioner’s contentions for this subject matter. *See generally* PO Resp. 41–54 (addressing “idle time period” subject matter).

We are persuaded by Petitioner’s contentions that Eggert teaches this “modifying” operation. In particular, Eggert discloses that, “[u]pon receipt of a segment with the Abort Timeout Option, the receiving host decides

IPR2020-00845

Patent 9,923,995 B1

whether to accept, shorten, or reject its peer’s proposed abort timeout.”

Ex. 1004, 5. Thus, we find that Eggert’s disclosure of shortening the proposed abort timeout teaches “modifying, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection,” as recited in claim 1.

d) Determination of unpatentability

For the reasons discussed above and based on the full trial record, we conclude that Petitioner has demonstrated by a preponderance of the evidence that the subject matter recited in claim 1 would have been obvious over the teachings of Eggert.

3. Independent Claim 15

Independent claim 15 recites “[a]n apparatus comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions.” Petitioner refers to its contentions for the same subject matter recited in claim 1. Pet. 51. For the reasons explained above in § II.D.2, we are persuaded that Eggert renders this subject matter obvious.

Claim 15 recites three operations, the first one being “receiving idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active.” Thus, claim 15 also recites the ITP subject matter. Petitioner argues that Eggert’s disclosure of the exchange of ATO information during the three-way handshake teaches this subject matter, similar to its contentions for the ITP limitation in claim 1, discussed above in § II.D.2.b. Pet. 51–54. Patent Owner argues claims 1 and 15 together, focusing on the ITP subject matter that is common to both.

PO Resp. 41–48. As discussed above in § II.D.2.b, we are persuaded by Petitioner’s contentions that the ITP subject matter would have been obvious over the teachings of Eggert. For the reasons given in the Petition at pages 51–54 and for those discussed above in § II.D.2.b, we are persuaded that Eggert’s disclosure renders obvious “receiving idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active.”

The second operation recited in claim 15 is “generating a TCP-variant packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information.” Petitioner argues that Eggert discloses generating packets having the ATO value, thereby teaching this subject matter. Pet. 54–56 (citing Ex. 1004, 3, 5, 9, Figs. 1, 2; Ex. 1003 ¶¶ 444–445).

During the trial, Patent Owner did not dispute Petitioner’s contentions for this subject matter. *See generally* PO Resp. 41–48 (addressing “idle time period” subject matter of claims 1 and 15).

We are persuaded by Petitioner’s contentions that Eggert teaches this “generating” operation. For example, Eggert discloses that the ATO value is included in any segment (packet) that has “a SYN flag, i.e., either the initial SYN or the SYN-ACK.” Ex. 1004, 5. Eggert’s Figure 2, cited by Petitioner (Pet. 54–55), illustrates “Allowed TCP Abort Timeout Option (ATO) Exchanges” and shows the ATO included in SYN and SYN/ACK packets. Ex. 1004, 7. For the reasons discussed above in § II.D.2.a, we find that these packets containing the ATO value are TCP-variant packets. *See* Pet. 56. Thus, we find that Eggert teaches “generating a TCP-variant packet

IPR2020-00845

Patent 9,923,995 B1

including an idle time period parameter field identifying metadata for the idle time period based on the idle information.”

The third operation recited in claim 15 is

sending, from a first node to a second node, the TCP-variant packet in advance of the TCP-variant connection being established to provide the metadata for the idle time period to the second node, for use by the second node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.

Petitioner contends that Eggert’s disclosure of exchanging packets with the ATO value between nodes and accepting or shortening the abort timeout attribute based on the exchanged ATO value teaches this subject matter.

Pet. 56–58 (citing Ex. 1004, 5, 7; Ex. 1003 ¶¶ 446–462).

During the trial, Patent Owner did not dispute Petitioner’s contentions for this subject matter. *See generally* PO Resp. 41–48 (addressing “idle time period” subject matter of claims 1 and 15).

We are persuaded by Petitioner’s contentions that Eggert teaches this “sending” operation. In particular, Eggert discloses that, “[u]pon receipt of a segment with the Abort Timeout Option, the receiving host decides whether to accept, shorten, or reject its peer’s proposed abort timeout.” Ex. 1004, 5; *see also* § II.D.2.c above (addressing “modifying, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection,” as recited in claim 1). Thus, we find that Eggert’s disclosure of exchanging proposed abort timeout values and shortening a proposed abort timeout teaches the “sending” operation recited in claim 15.

For the reasons discussed above and based on the full trial record, we conclude that Petitioner has demonstrated by a preponderance of the

evidence that the subject matter recited in claim 15 would have been obvious over the teachings of Eggert.

4. Dependent Claims 11 and 21

Dependent claims 11 and 21 recite some similar subject matter, but claim 11 includes an additional recitation as to which Patent Owner specifically challenges Petitioner's showing. We discuss the overlapping subject matter of these claims in subsection a) and the additional recitation of claim 11 in subsection b) below.

a) Detecting the idle time period and deactivating the connection

Claim 11 depends from claim 1 and recites “wherein the one or more processors execute the instructions for: detecting the idle time period based on the timeout attribute; and in response to detecting the idle time period, deactivating the TCP-variant connection by releasing a resource allocated for the TCP-variant connection by one of the first and second nodes.”

Claim 21 depends from claim 15 and recites “wherein the one or more processors execute the instructions for: detecting the idle time period based on the idle information; and in response to detecting the idle time period, deactivating the TCP-variant connection.”

Petitioner argues that “Eggert discloses that the host will use the negotiated abort timeout, rather than the default user timeout, to detect whether the idle time period, i.e., how long segments remain unacknowledged, has expired.” Pet. 42 (citing Ex. 1004, 3). Petitioner contends that a person of ordinary skill in the art “would find it obvious to implement Eggert with a node that would execute instructions to measure the amount of time that segments remain unacknowledged and detect when the time elapsed reaches or exceeds the negotiated abort timeout.” Pet. 43 (citing Ex. 1003 ¶¶ 510–512). Petitioner argues that Eggert teaches

IPR2020-00845

Patent 9,923,995 B1

deactivating the connection by releasing a resource with its disclosure that “[i]f a disconnection lasts longer than the user timeout, the TCP connection will abort.” Pet. 43 (quoting Ex. 1004, 3).

Patent Owner does not dispute Petitioner’s contentions for this subject matter. *See* PO Resp. 48–54 (addressing different recitation of claim 11).

We are persuaded by Petitioner’s contentions that it would have been obvious for a node to detect the time period defined by the negotiated ATO because Eggert specifies that the “connection will abort” when the “disconnection lasts longer than the user timeout.” Ex. 1004, 3. Based on Petitioner’s persuasive contentions and evidence, therefore, we are persuaded that the above-quoted subject matter of claims 11 and 21 would have been obvious to a person of ordinary skill in the art based on Eggert’s disclosure.

b) Deactivating without signaling

Claim 11 further specifies that the deactivating occurs “without signaling another one of the first and second nodes that is related to the detection of the idle time period.” Petitioner argues that “[t]he ATO of Eggert does not specify any mechanism to signal another node that an idle time period has been detected.” Pet. 43. Petitioner further argues that a person of ordinary skill in the art would not implement Eggert to signal the other node because “the Eggert system assumes a disconnection occurred so the signal may not reach the other node.” Pet. 44 (citing Ex. 1003 ¶¶ 375–377; Ex. 1006, 36–38, 40¹¹).

¹¹ The Petition cites the page numbers that appear as part of RFC 793 itself, which appear in brackets at the bottom of every page. We cite the exhibit page numbers in the lower right-hand corner of Exhibit 1006.

IPR2020-00845

Patent 9,923,995 B1

Patent Owner responds that Eggert's disclosure that the "connection will abort" (Ex. 1004, 3) means that a signal is sent to the other side because RFC 793 defines the "Abort" command as including the sending of a RESET message. PO Resp. 48–50 (citing Ex. 1006, 54; Ex. 2020 ¶¶ 88–104). Patent Owner also cites disclosures in RFC 793 about the sending of RESET messages in certain circumstances. PO Resp. 51–52 (citing Ex. 1006, 37–39, Fig. 10; Ex. 2020 ¶¶ 90–104).

For the reasons explained below, we are persuaded by Petitioner's contentions that the subject matter reciting "without signaling" would have been obvious in view of Eggert. Eggert discloses the following: "The TCP specification includes a 'user timeout' that defines the maximum amount of time that segments may remain unacknowledged before TCP will abort the connection. If a disconnection lasts longer than the user timeout, the TCP connection will abort." Ex. 1004, 3 (citing RFC 793 (Ex. 1006 in the record)); *see also* Ex. 2020 ¶ 30 ("*Eggert's* ATO utilizes the TCP specification in which there is a 'user timeout' that defines the amount of time that packets may remain unacknowledged before the connection is terminated. (Ex. 1004 at 3.) The failure to *receive* an ACK packet triggers termination of the connection. (*Id.*)"). Although Eggert uses the term "abort," the TCP Specification (RFC 793) provides the following description of a "USER TIMEOUT": "For any state if the user timeout expires, flush all queues, signal the user 'error: connection aborted due to user timeout' in general and for any outstanding calls, delete the TCB, enter the CLOSED state and return." Ex. 1006, 81, *quoted in* Pet. 44. RFC 793 further explains that "CLOSED is fictional because it represents the state when there is no [Transmission Control Block (TCB)], and therefore, no connection." Ex. 1006, 25; *see also* Ex. 1006, 26 ("CLOSED - represents no connection

IPR2020-00845

Patent 9,923,995 B1

state at all.”). Based on these disclosures, we agree with Petitioner that “the RESET message is not sent when the user timeout expires.” *See* Pet. 44 (citing Ex. 1006, 77).

Although “a special RESET message” is sent as part of the ABORT command, this is a TCP user command or user call, as Petitioner correctly points out. *See* Pet. Reply 24–25 (citing Ex. 1006, 48, 54, 56). RFC 793 states that “[t]he activity of the TCP can be characterized as responding to events,” which “can be cast into three categories: user calls, arriving segments, and timeouts.” Ex. 1006, 56, *cited in* Pet. Reply 25. RFC 793 categorizes “ABORT” as a “User Call[]” event and “USER TIMEOUT” as a “Timeout[]” event. Ex. 1006, 56, *cited in* Pet. Reply 25. Thus, we find that Eggert’s disclosure of the “user timeout” in the “TCP Specification” (RFC 793) refers to the timeout event in RFC 793, in which no reset message is sent. Ex. 1004, 81 (“For any state if the user timeout expires, flush all queues, signal the user ‘error: connection aborted due to user timeout’ in general and for any outstanding calls, delete the TCB, enter the CLOSED state and return.”).

Based on the foregoing, we find that Eggert teaches deactivating the connection “without signaling another one of the first and second nodes that is related to the detection of the idle time period,” as recited in claim 11.

c) Determination of unpatentability

For the reasons discussed above and based on the full trial record, we conclude that Petitioner has demonstrated by a preponderance of the evidence that the subject matter recited in claims 11 and 21 would have been obvious over the teachings of Eggert.

IPR2020-00845

Patent 9,923,995 B1

5. *Dependent Claims 2–10, 12–14, 16–20, and 22–24*

Petitioner contends that dependent claims 2–10, 12–14, 16–20, and 22–24 are unpatentable as obvious over the teachings of Eggert. Pet. 35–42, 45–51, 58–61. For these dependent claims, Patent Owner refers to its arguments regarding the ITP subject matter of the independent claims and does not set forth additional arguments for the particular subject matter recited in these claims. *See* PO Resp. 55. As explained below, we are persuaded by Petitioner’s contentions for dependent claims 2–10, 12–14, 16–20, and 22–24.

Claim 2 recites, “The apparatus of claim 1 wherein the apparatus is configured such that the timeout attribute is an attribute of a keep-alive.” Citing Eggert’s disclosure that the ATO “allows mobile hosts to maintain TCP connections across disconnected periods” (Ex. 1004, 3), Petitioner argues that “the initiator and responder in Eggert keep the TCP connection alive based on information contained in the ATO.” Pet. 35–36 (citing Ex. 1004, Abstract, 3; Ex. 1003 ¶¶ 271–277).

Claim 3 recites, “The apparatus of claim 1 wherein at least one of: the second node includes a server . . . or the second node includes a client,” and each alternative recites an operation for which the server or client is configured. For example, the first alternative recites “the server being configured to: in response to the receiving, send, by the second node to the first node, another TCP-variant packet in advance of the TCP-variant connection being established, the another TCP-variant packet including other metadata for the idle time period.” Petitioner argues that Eggert’s disclosure of the responder sending a SYN/ACK+ATO segment teaches this subject matter. Pet. 36–37 (citing Ex. 1004, 5–7, 11–13, Fig. 2; Ex. 1003 ¶¶ 279–291). Petitioner also argues that Eggert teaches the second

IPR2020-00845

Patent 9,923,995 B1

alternative recited in claim 3. Pet. 37–38 (citing Ex. 1004, 1–5, Fig. 2; Ex. 1003 ¶¶ 192, 212, 221, 292–298).

Claim 4 depends from claim 3 and recites “wherein, regardless as to whether the apparatus is the server or the client, the metadata is the same as the other metadata.” Claim 5 depends from claim 3 and recites “wherein, regardless as to whether the apparatus is the server or the client, the metadata is different from the other metadata.” Petitioner argues that Eggert teaches the subject matter in both of these claims because the receiving host can either accept the ATO value it receives (“same” as recited in claim 4) or shorten it (“different” as recited in claim 5). Pet. 38–39 (citing Ex. 1004, 5–11; Ex. 1003 ¶¶ 299–316).

Claim 6 depends from claim 1 and recites that “the timeout attribute is specified in at least one of a number of seconds or minutes,” and claim 19 depends from claim 15 and recites that “the timeout attribute is specified in a number of seconds.” Petitioner cites Eggert’s disclosure that “the Abort Timeout Option specifies abort timeouts as 32-bit values with a granularity of seconds.” Pet. 39 (quoting Ex. 1004, 11) (citing Ex. 1003 ¶¶ 317–322).

Claim 7 depends from claim 1 and recites that “the apparatus is configured such that the timeout attribute is used to keep the TCP-variant connection open when inactive, and to prevent one or more other nodes from closing the TCP-variant connection when inactive.” Citing Eggert’s disclosure that the ATO allows nodes “to maintain TCP connections across disconnected periods that are longer than their system’s default abort timeout” (Ex. 1004, 3), Petitioner argues that Eggert’s ATO keeps the connection open when inactive and prevents closure of the connection, thereby teaching the subject matter of claim 7. Pet. 39–40 (citing Ex. 1004, Abstract, 3, 9; Ex. 1003 ¶¶ 327–332).

IPR2020-00845

Patent 9,923,995 B1

Claim 8 depends from claim 1 and recites “wherein the apparatus is configured such that the metadata is used as input of an algorithm for determining a duration of time specified by the timeout attribute.” Petitioner argues that “[i]n Eggert, the receiving host performs an algorithm to decide whether to shorten, accept, or reject the proposed ATO,” and Petitioner cites several disclosures in Eggert as examples of algorithms for determining the timeout duration. Pet. 40 (citing Ex. 1004, 5, 9–13; Ex. 1003 ¶¶ 333–344). For example, Eggert discloses that “implementations can require prior peer authentication . . . before accepting long abort timeouts for the peer’s connections.” Ex. 1004, 13.

Claim 9 depends from claim 8 and recites that “the apparatus is configured such that the algorithm is determined based on at least one particular attribute.” Petitioner argues that “the algorithm may be determined based on attributes, including a security feature or a confirmation of prior peer authentication.” Pet. 41 (citing Ex. 1004, 13). As discussed in the preceding paragraph for claim 8, Eggert discloses an algorithm involving “prior peer authentication.” Ex. 1004, 13.

Claim 10 depends from claim 1 and recites that “the apparatus is configured such that the modification of the timeout attribute results from a negotiation between the first node and the second node via a negotiation protocol of a TCP-variant protocol.” Claim 20 depends from claim 15 and recites that “the apparatus is configured such that the modification of the timeout attribute results from a negotiation between the first node and the second node.” Petitioner argues that this subject matter is taught by Eggert because the “abort timeout value is negotiated during the connection’s three-way handshake.” Pet. 41–42 (citing Ex. 1004, 5, 9, Fig. 2; Ex. 1003 ¶¶ 355–362, 500–504).

IPR2020-00845

Patent 9,923,995 B1

Claim 12 depends from claim 1 and recites “wherein the apparatus is configured such that at least one of the detecting or the identifying is performed at a TCP-variant layer other than a TCP layer, where the TCP-variant layer is above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer.” Claim 22 depends from claim 15 and recites “wherein the apparatus is configured such that at least the generating is performed at a TCP-variant layer other than a TCP layer, where the TCP-variant layer is above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer.” Citing the testimony of its declarant, Mr. Scott Bradner, Petitioner argues that “HTTP applications are application layer protocols that run above transport layer protocols (e.g., TCP), which run above the network layer (e.g., IP layer).” Pet. 46 (citing Ex. 1003 ¶¶ 390–392, 522–523). According to Petitioner, therefore, Eggert’s TCP-variant protocol would be implemented in a TCP-variant layer, which a person of ordinary skill in the art would have implemented “between the application layer (e.g., HTTP) and the IP layer, just like the original TCP layer.” Pet. 45–46 (citing Ex. 1004, 5; Ex. 1003 ¶¶ 381–392, 516–523).

Claim 13 depends from claim 1 and recites

wherein the one or more processors execute a network application that is configured to perform a 3-way TCP handshake for establishing a TCP connection that is different than the TCP-variant connection, and wherein the network application is configured to establish the TCP-variant connection instead of the TCP connection in order to permit negotiation, between the first node and the second node, of the timeout attribute, where the timeout attribute is not available when establishing the TCP connection, but is available when establishing the TCP-variant connection so that the TCP-variant connection is capable of

IPR2020-00845

Patent 9,923,995 B1

being at least partially closed when inactive based on the timeout attribute.

Claim 23 depends from claim 15 and recites similar subject matter.

With regard to these claims, Petitioner argues that Eggert's ATO is optional and that a person of ordinary skill in the art "would have found it obvious to implement Eggert with a host, for example (e.g., a server), that supports both the standard TCP and Eggert's TCP modification." Pet. 46–47 (citing Ex. 1004, 7–13; Ex. 1003 ¶¶ 396–403, 527–528). For example, Eggert discloses that "[a] TCP implementation that does not support the TCP Abort Timeout Option SHOULD silently ignore it," which "causes the connection to use the default abort timeout, thus ensuring interoperability." Ex. 1004, 7. Petitioner further argues that "a host that wishes to negotiate the timeout must resort to exchange Eggert's ATO parameters during the three-way handshake," thereby teaching negotiation of the timeout attribute. Pet. 48 (citing Ex. 1004, 3–5; Ex. 1003 ¶¶ 404–409). Petitioner argues that "[t]he negotiated abort timeout attribute of Eggert is not available when establishing the TCP connection but is available when establishing the Eggert TCP-variant connection." Pet. 49 (citing Ex. 1004, 9–13). Finally, Petitioner argues that "Eggert's TCP-variant connection is fully closed, meeting the limitation of being 'at least partially closed,' when the host determines that the connection is inactive based on the abort timeout." Pet. 50 (citing Ex. 1003 ¶¶ 417–418, 539).

Claim 14 depends from claim 1 and recites "wherein the apparatus is configured such that the TCP-variant packet and the metadata included therewith are received by the second node from the first node, without any prior signaling from the second node to the first node." Claim 17 depends from claim 15 and recites "wherein the apparatus is configured such that the

IPR2020-00845

Patent 9,923,995 B1

TCP-variant packet and the metadata included therewith are sent by the first node to the second node, without any prior signaling received by the first node from the second node.” Petitioner argues that Eggert teaches this subject matter because “[t]he messages exchanged in the Eggert three-way handshake are the first messages received by any node to establish the Eggert TCP-variant connection.” Pet. 51 (citing Ex. 1004, Fig. 2; Ex. 1003 ¶¶ 423–425); *see* Pet. 59 (addressing claim 17). In support, Petitioner produces annotated excerpts from Eggert’s Figure 2 (not reproduced here) noting that “SYN+ATO” is the first signal with no prior signaling. Pet. 51, 59.

Claim 16 depends from claim 15 and recites “wherein the apparatus is configured such that modifying the timeout attribute reduces a number of keep-alive signals that are required to be communicated.” Petitioner cites keep-alive messages in RFC 1122, which Eggert cites as a normative reference (Ex. 1004, 15), and argues that “[o]ne obvious implementation of Eggert is employing keep-alive packets per RFC 1122.” Pet. 58 (citing Ex. 1004, 3, 15; Ex. 1003 ¶¶ 151, 476; Ex. 1007). According to Petitioner, “[s]ince the keep-alive packets are sent at a regular interval, shortening the abort timeout will reduce the number of keep-alive signals that are transmitted when there is a disconnection.” Pet. 58–59.

Claim 18 depends from claim 15 and recites “wherein the metadata includes a first value and the timeout attribute is capable of being modified, by the second node, to include a second value that is different than the first value of the metadata.” Petitioner argues that, in Eggert, the metadata includes the ATO value and that this value can be shortened, i.e., “modified” during the three-way handshake negotiation. Pet. 59–60 (citing Ex. 1004, 5, 11, Figs. 1, 2; Ex. 1003 ¶¶ 490–494). Eggert discloses that, “[u]pon receipt

IPR2020-00845

Patent 9,923,995 B1

of a segment with the Abort Timeout Option, the receiving host decides whether to accept, *shorten*, or reject its peer's proposed abort timeout."

Ex. 1004, 5 (emphasis added).

Claim 24 depends from claim 15 and recites "wherein the apparatus is configured such that the TCP-variant packet and the metadata included therewith are sent by the first node to the second node, in response to receiving, by the first node from the second node, another TCP-variant packet with other metadata." Petitioner argues that Eggert's "SYN/ACK+ATO segment from the responder (first node) is a TCP-variant packet that contains in the ATO sub-field metadata for an abort timeout to be used for the connection and is a response to a first SYN+ATO segment received from the initiator (second node)." Pet. 61 (citing Ex. 1003 ¶¶ 541–546).

As noted at the beginning of this section, Patent Owner does not separately address the additional subject matter recited in these dependent claims. *See generally* PO Resp.

Having considered the full record developed during trial, we are persuaded by Petitioner's contentions and evidence, summarized above, and we determine that Petitioner has proven by a preponderance of the evidence that dependent claims 2–10, 12–14, 16–20, and 22–24 are unpatentable as obvious over the teachings of Eggert.

E. Remaining Grounds

As noted above in § I.A, Petitioner presents separate challenges to the claims in this proceeding. Because we determine that all challenged claims are unpatentable as discussed above, we need not separately assess the patentability of these claims based on the remaining grounds in this proceeding. 35 U.S.C. § 318(a) ("If an inter partes review is instituted and

IPR2020-00845

Patent 9,923,995 B1

not dismissed under this chapter, the Patent Trial and Appeal Board shall issue a final written decision with respect to the patentability of any patent claim challenged by the petitioner and any new claim added under section 316(d).”); *Bos. Sci. Scimed, Inc. v. Cook Grp. Inc.*, 809 F. App’x 984, 990 (Fed. Cir. 2020) (nonprecedential) (“We agree that the Board need not address issues that are not necessary to the resolution of the proceeding.”).

III. CONCLUSION¹²

For the reasons discussed above, we determine that Petitioner has proven, by a preponderance of the evidence, that the challenged claims are unpatentable, as summarized in the following table:

Claim(s)	35 U.S.C. §	Reference(s)/Basis	Claims Shown Unpatentable	Claims Not Shown Unpatentable
1–24	103	Eggert	1–24	
1–24	103	Eggert, Hankinson ¹³		
16	103	Eggert, Hankinson, RFC 1122		
Overall Outcome			1–24	

¹² Should Patent Owner wish to pursue amendment of the challenged claims in a reissue or reexamination proceeding subsequent to the issuance of this decision, we draw Patent Owner’s attention to the April 2019 *Notice Regarding Options for Amendments by Patent Owner Through Reissue or Reexamination During a Pending AIA Trial Proceeding*. See 84 Fed. Reg. 16,654 (Apr. 22, 2019). If Patent Owner chooses to file a reissue application or a request for reexamination of the challenged patent, we remind Patent Owner of its continuing obligation to notify the Board of any such related matters in updated mandatory notices. See 37 C.F.R. § 42.8(a)(3), (b)(2).

¹³ As explained above, because we determine that the challenged claims are unpatentable on another ground, we decline to address the remaining grounds.

IPR2020-00845

Patent 9,923,995 B1

IV. ORDER

Accordingly, it is

ORDERED that claims 1–24 of the '995 patent have been shown to be unpatentable; and

FURTHER ORDERED that, because this is a Final Written Decision, parties to the proceeding seeking judicial review of the Decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

IPR2020-00845

Patent 9,923,995 B1

FOR PETITIONER:

Naveen Modi

Joseph E. Palys

Quadeer A. Ahmed

Jason Heidemann

Ph-google-jenam-IPR@paulhastings.com

naveenmodi@paulhastings.com

josephpalys@paulhastings.com

quadeerahmed@paulhastings.com

jasonheidemann@paulhastings.com

FOR PATENT OWNER:

Timothy Devlin

Derek Dahlgren

DEVLIN LAW FIRM LLC

td-ptab@devlinlawfirm.com

ddahlgren@devlinlawfirm.com

EXHIBIT 13

From: [Derek Dahlgren](#)
To: [Matthew Werdegarg](#); [Jason George](#)
Cc: [tdevlin@devlinlawfirm.com](#); [scheruvu@devlinlawfirm.com](#); [dlflitparas@devlinlawfirm.com](#); [JENAMG](#)
Subject: RE: Jenam Tech, LLC v. Google LLC - Asserted Claims
Date: Tuesday, August 31, 2021 6:18:55 PM

[EXTERNAL]

Matt,

We are still asserting the claims from the '995 patent. We also want to make sure Google is aware that for the '995 patent, it is estopped from asserting prior art defenses based on estoppel under § 315(e)(2). Please confirm that Google will not assert any such prior art defenses.

Best regards,
Derek

From: Matthew Werdegarg <mwerdegarg@keker.com>
Sent: Monday, August 30, 2021 9:22 PM
To: Derek Dahlgren <ddahlgren@devlinlawfirm.com>; jgeorge@keker.com
Cc: Timothy Devlin <tdevlin@devlinlawfirm.com>; Srikant Cheruvu <scheruvu@devlinlawfirm.com>; DLF-Lit Paras <dlflitparas@devlinlawfirm.com>; jenamg@keker.com
Subject: RE: Jenam Tech, LLC v. Google LLC - Asserted Claims

Derek,

In light of the PTAB's Judgment and Final Written Decision in the '995 IPR invalidating claims 1-24 of the '995 patent, please confirm that Jenam is no longer going to assert any '995 patent claims in the district court litigation and that Google will not have to address any '995 patent claims in forthcoming invalidity contentions and claim construction proceedings.

Thank you,
Matt

Matthew Werdegarg
Keker, Van Nest & Peters LLP
633 Battery Street
San Francisco, CA 94111-1890
415 676 2248 direct | 415 391 5400 main
mwerdegarg@keker.com | [vcard](#) | keker.com
Pronouns: he / him

From: Derek Dahlgren <ddahlgren@devlinlawfirm.com>
Sent: Friday, August 20, 2021 4:14 PM
To: Jason George <JGeorge@keker.com>
Cc: tdevlin@devlinlawfirm.com; scheruvu@devlinlawfirm.com; dlflitparas@devlinlawfirm.com; JENAMG <JENAMG@keker.com>
Subject: RE: Jenam Tech, LLC v. Google LLC - Asserted Claims

[EXTERNAL]

Jason,

I've confirmed that any asserted claim which depends from one of the disclaimed claims is still being asserted. Please let me know if you have any additional questions.

Best regards,
Derek

From: Jason George <jgeorge@keker.com>
Sent: Friday, August 20, 2021 2:00 PM
To: Derek Dahlgren <ddahlgren@devlinlawfirm.com>
Cc: Timothy Devlin <tdevlin@devlinlawfirm.com>; Srikant Cheruvu <scheruvu@devlinlawfirm.com>; DLF-Lit Paras <dlflitparas@devlinlawfirm.com>; jenamg@keker.com
Subject: RE: Jenam Tech, LLC v. Google LLC - Asserted Claims

Thanks, Derek. Please let us know which claims are no longer asserted against Google based on the disclaimers.

Best,
Jason

Jason George
Keker, Van Nest & Peters LLP
633 Battery Street
San Francisco, CA 94111-1809
415 962 8847 direct | 415 391 5400 main
jgeorge@keker.com | [vcard](#) | keker.com
Pronouns: he/him/his

From: Derek Dahlgren <ddahlgren@devlinlawfirm.com>
Sent: Thursday, August 19, 2021 6:17 PM

To: Jason George <JGeorge@keker.com>

Cc: tdevlin@devlinlawfirm.com; scheruvu@devlinlawfirm.com; dlflitparas@devlinlawfirm.com;

JENAMG <JENAMG@keker.com>

Subject: RE: Jenam Tech, LLC v. Google LLC - Asserted Claims

[EXTERNAL]

Jason, I believe that is correct for the specific claims that were disclaimed, but not for any asserted claims that depend from them. I don't have that information in front of me at the moment, but I'll confirm tomorrow after reviewing the specific claims and contentions.

From: Jason George <jgeorge@keker.com>

Sent: Thursday, August 19, 2021 9:14 PM

To: Derek Dahlgren <ddahlgren@devlinlawfirm.com>

Cc: Timothy Devlin <tdevlin@devlinlawfirm.com>; Srikant Cheruvu <scheruvu@devlinlawfirm.com>;

DLF-Lit Paras <dlflitparas@devlinlawfirm.com>; jenamg@keker.com

Subject: Jenam Tech, LLC v. Google LLC - Asserted Claims

Derek,

Given that Jenam disclaimed claims 16, 22, 24, 28, and 29 of the '564 patent, claim 25 of the '565 patent, and claims 52, 61, 68, and 69 of the '026 patent in the parallel IPR Proceedings (IPR 2021-00628 ('564 patent); IPR 2021-00629 ('564 patent); IPR2021-00630 ('565 patent); IPR2021-00868 ('026 patent)), please confirm that Jenam is no longer asserting those claims or claims that depend from them against Google.

Thanks,
Jason

Jason George

Keker, Van Nest & Peters LLP

633 Battery Street

San Francisco, CA 94111-1809

415 962 8847 direct | 415 391 5400 main

jgeorge@keker.com | [vcard](#) | keker.com

Pronouns: he/him/his

EXHIBIT 14

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

JENAM TECH, LLC,
Patent Owner.

IPR2021-00628
Patent 10,075,564 B1

Before DANIEL J. GALLIGAN, SCOTT B. HOWARD, and
JASON M. REPKO, *Administrative Patent Judges*.

GALLIGAN, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

IPR2021-00628

Patent 10,075,564 B1

I. INTRODUCTION

A. Background

Google LLC (“Petitioner”) filed a Petition requesting *inter partes* review of claims 16, 22, and 23 of U.S. Patent No. 10,075,564 B1 (“the ’564 patent,” Ex. 1001). Paper 1 (“Pet.”). Jenam Tech, LLC (“Patent Owner”) filed a Preliminary Response, in which Patent Owner stated that it had “statutorily disclaimed claims 16 and 22,” leaving only claim 23 for consideration. Paper 7 (“Prelim. Resp.”), 1.

With our authorization, Petitioner and Patent Owner filed, respectively, a Reply (Paper 9) and a Sur-reply (Paper 10) addressing discretionary denial issues.

Under 37 C.F.R. § 42.4(a), we have authority to determine whether to institute review. The standard for instituting an *inter partes* review is set forth in 35 U.S.C. § 314(a), which provides that an *inter partes* review may not be instituted unless the information presented in the Petition and the Preliminary Response shows “there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.”

For the reasons explained below, we institute an *inter partes* review as to challenged claim 23 on the sole ground raised in the Petition.

B. Related Matters

As required by 37 C.F.R. § 42.8(b)(2), the parties identify various related matters, including another petition for *inter partes* review challenging different claims of the ’564 patent. Pet. 1–2; Paper 5; *see also* IPR2021-00629, Paper 1 (challenging claims 24–29 of the ’564 patent).

IPR2021-00628

Patent 10,075,564 B1

C. Real Parties in Interest

The parties identify themselves as the real parties in interest. Pet. 1; Paper 5.

D. The '564 Patent and Illustrative Claims

The '564 patent states that some transmission control protocol (TCP) implementations support a keep-alive option, but the '564 patent states that “[e]ach node supports or does not support keep-alive for a TCP connection based on each node’s requirements without consideration for the other node in the TCP connection.” Ex. 1001, 1:48–50, 1:58–61. According to the '564 patent, “[t]o date no mechanism to allow two TCP connection endpoints to cooperate in supporting the keep-alive option has been proposed or implemented.” Ex. 1001, 2:19–22. Thus, the '564 patent states that “there exists a need for methods, systems, and computer program products for sharing information for detecting an idle TCP connection.” Ex. 1001, 2:26–28.

Claim 23, the only remaining claim, depends from claims 16 and 22. All three claims are reproduced below.

16. A non-transitory computer readable medium, comprising:
code for being communicated to a remote client node including one or more processors in communication with a non-transitory memory, where the code, when used by the client node, results in the client node operating to:
 - identify idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active;
 - generate a TCP-variant packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information; and
 - send, from the client node to a server node, the TCP-variant packet to provide the metadata for the idle time

IPR2021-00628

Patent 10,075,564 B1

period to the server node, for use by the server node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.

22. A system including the non-transitory computer readable medium of claim 16, wherein the system: includes another server node separate from the server node, is configured to communicate the code to the client node for use in establishing the TCP-variant connection with the server node, and is further configured to perform a 3-way TCP handshake for establishing, with the client node, a TCP connection that is different than the TCP-variant connection for permitting the client node to fetch the code in addition to other data from the system via a hypertext transfer protocol (HTTP).

23. The system of claim 22, wherein the code, when used by the client node, results in the client node operating such that the TCP-variant connection is established between the client node and the server node instead of the TCP connection in order to permit communication, between the client node and the server node, of the timeout attribute, where the timeout attribute is not communicated when establishing the TCP connection between the system and the client node, but is communicated when establishing the TCP-variant connection between the client and the server node.

E. Asserted Ground of Unpatentability

Petitioner challenges the patentability of claim 23 of the '564 patent on the following ground:

Claim(s) Challenged	35 U.S.C. §	Reference(s)/Basis
23	103	Wookey, ¹ Eggert ²

¹ Wookey, US 2007/0171921 A1, published July 26, 2007 (Ex. 1005).

² L. Eggert, "TCP Abort Timeout Option," Apr. 14, 2004 (Ex. 1006).

IPR2021-00628

Patent 10,075,564 B1

II. ANALYSIS

A. Discretionary Denial

1. 35 U.S.C. § 314(a)³

Patent Owner argues that we should exercise discretion to deny the Petition under 35 U.S.C. § 314(a) based on the factors set forth in *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 (PTAB Sept. 6, 2017) (designated precedential in relevant part). Prelim. Resp. 57–65. Patent Owner cites other petitions for *inter partes* review that have been filed challenging claims of Patent Owner’s patents and argues that, “[b]ecause [Petitioner’s] petitions challenge related patents having overlapping claim limitations that are directed to the same subject, they place an unfair burden on [Patent Owner] and should be denied.” Prelim. Resp. 59–60.

We do not agree with Patent Owner’s argument that this is the type of “follow-on” petition that should be denied under the Board’s precedential *General Plastic* decision. *General Plastic* sets forth a series of factors to be considered by the Board in evaluating whether to exercise discretion under § 314(a) to deny a petition that challenges a patent that was previously challenged before the Board. *General Plastic*, Paper 19 at 15–19. As *General Plastic* states, “[m]ultiple, staggered petitions *challenging the same patent and same claims* raise the potential for abuse.” *Id.* at 17 (emphasis added). Here, Patent Owner cites numerous petitions against related patents but only one other petition challenging claims of the ’564 patent. Prelim.

³ Although the Preliminary Response identifies both sections 314(a) and 325(d) in the heading, the body of the argument only discusses section 314(a). See Prelim. Resp. 57–65. Accordingly, we only address section 314(a) here.

IPR2021-00628

Patent 10,075,564 B1

Resp. 59–60. The other petition challenging the '564 patent (IPR2021-00629, Paper 1) was filed the same date as the present Petition and challenges different claims of the '564 patent. Prelim. Resp. 59; Paper 4 (according filing date of March 15, 2021); IPR2021-00629, Paper 4 (according filing date of March 15, 2021). Thus, this is not a situation involving “[m]ultiple, staggered petitions challenging the same patent and same claims” contemplated by *General Plastic*. See *General Plastic*, Paper 19 at 17. Rather, this is a situation involving parallel petitions, which we address below in § II.A.2.

Patent Owner argues that “[t]he Board has denied institution of IPRs filed against related patents because it placed an unfair burden on the patent owner.” Prelim. Resp. 58 (citing *Tietex Int’l, LTD., v. Precision Fabrics Group, Inc.*, IPR2015-01671, Paper 7, at 8 (PTAB Feb. 11, 2016)).

The decision denying institution in *Tietex*, which is not precedential, is readily distinguishable from the present Petition. In *Tietex*, the panel denied institution on one ground⁴ where the petition presented “the same arguments that we[re] found to be unpersuasive in” a previous final written decision addressing claims of similar scope in a related patent. *Tietex*, Paper 7 at 8. By contrast, we have issued a final written decision in IPR2020-00845 holding all challenged claims of U.S. Patent 9,923,995 B1 unpatentable. See *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845, Paper 33 (PTAB Aug. 27, 2021). Therefore, the reasoning in *Tietex* does not apply to the facts of the present case.

⁴ The *Tietex* decision issued before the Supreme Court’s holding that a decision to institute under 35 U.S.C. § 314 may not institute on fewer than all claims challenged in the petition. See *SAS Inst., Inc. v. Iancu*, 138 S. Ct. 1348, 1359–60 (2018).

IPR2021-00628

Patent 10,075,564 B1

For the foregoing reasons, we do not deny the Petition based on *General Plastic* under § 314(a).

2. *Multiple Petitions*

Patent Owner argues that “the November 2019 Consolidated Trial Practice Guide states that petitioners that file multiple petitions against the same patent must submit” a ranking of the petitions and an explanation of the differences. Prelim. Resp. 66 (citing Consolidated Trial Practice Guide (Nov. 2019), 59–60). Patent Owner argues that Petitioner “failed to submit this information” and, therefore, that the Petition should be denied. Prelim. Resp. 66.

Petitioner, however, did submit this information as Paper 3 in this proceeding. Petitioner explains that it filed different petitions to address different claim sets—one set (claims 16, 22, and 23) reciting “TCP-variant” and the other set (claims 24–29) reciting “non-TCP.” Paper 3, 2. Petitioner argues that,

[g]iven the difference in the protocol used between claims 16, 22, and 23 and claims 24–29 of the ’564 patent, as well as the numerous elements in each of the claims (including the dependent claims), Petitioner submits that challenging all claims of the ’564 patent in a single petition is not feasible or administratively efficient.

Paper 3, 2. As Patent Owner appears to have overlooked Petitioner’s explanation (*see* Prelim. Resp. 66), Patent Owner did not respond to Petitioner’s explanation in the Preliminary Response or in a separate paper, as it was entitled to do. *See* Consolidated Trial Practice Guide (Nov. 2019), 60–61 (authorizing a patent owner to file “[a] separate paper . . . with the preliminary responses . . . limited to providing this explanation” of “no more than 5 pages where the same paper is filed with each preliminary response”).

IPR2021-00628

Patent 10,075,564 B1

We find Petitioner’s explanation for filing two petitions challenging claims of the ’564 patent to be satisfactory. Each petition challenges different claims and presents only one ground against each claim. Paper 3, 1; Pet. 3 (challenging claims 16, 22, and 23 based on the combination of Wookey and Eggert); IPR2021-00629, Paper 1, 3 (challenging claims 24–29 based on the combination of Wookey and Berg⁵). Therefore, this is not a situation where a petitioner is using two petitions to lodge multiple parallel challenges to the same claims. Also, as Petitioner points out (Paper 3, 2), the two petitions challenge different claim sets that are directed to different protocols. Claims 24–29 recite “non-TCP connection” and “non-TCP packet,” whereas claims 16, 22, and 23 recite “TCP-variant connection” and “TCP-variant packet.” Petitioner relies on different references to account for these different protocols—Eggert for “TCP-variant” and Berg for “non-TCP.” Paper 3, 3–4.

For the above reasons, we decline to exercise our discretion to deny the Petition based on Petitioner’s filing two petitions challenging the ’564 patent.

B. Principles of Law

A patent claim is unpatentable under 35 U.S.C. § 103 if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art;

⁵ Berg, US 6,674,713 B1, issued Jan. 6, 2004 (IPR2021-00629, Ex. 1007).

IPR2021-00628

Patent 10,075,564 B1

(2) any differences between the claimed subject matter and the prior art;
 (3) the level of ordinary skill in the art; and (4) any secondary considerations, if in evidence.⁶ *Graham v. John Deere Co. of Kan. City*, 383 U.S. 1, 17–18 (1966).

C. Claim Construction

We interpret claim terms using “the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. § 42.100(b) (2019).

1. Idle Time Period (“ITP”)

Claim 16, from which claim 23 ultimately depends, recites “identify idle information for detecting an *idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active*” (emphasis added).

Patent Owner’s proposed construction for the claim recitation italicized above is in the following table:

Claim Language	Patent Owner’s Proposed Construction
“ <i>idle time period</i> , during which, <i>no packet is communicated</i> in a transmission control protocol (TCP)-variant connection <i>to keep the TCP-variant connection active</i> ”	“ <i>idle time period</i> and, during which, no packet <i>of any kind</i> is <i>sent or received by the [system]</i> in the TCP-variant connection to keep the TCP-variant connection active <i>with or without a condition and via any mechanism</i> ”

⁶ Patent Owner does not present any objective evidence of nonobviousness (i.e., secondary considerations) as to any of the challenged claims.

IPR2021-00628

Patent 10,075,564 B1

Prelim. Resp. 8–9; *see also id.* at 8–25 (argument concerning these constructions).

As set forth in the table above, Patent Owner’s proposed construction is based on the following three terms or phrases in the recited subject matter: “no packet,” “communicated,” and “to keep the TCP-variant connection active.” Prelim. Resp. 8–25. First, as to the language “no packet,” Patent Owner argues that there is no “modifier” for “no packet” and, therefore, that “the ‘no packet’ limitation is *unqualified*, prohibiting *all* types of packets from being communicated to keep the TCP-variant connection active.” Prelim. Resp. 10. Based on the current record, we disagree. Although the phrase “no packet” is absolute, the remainder of the claim includes a significant qualifier by reciting that no packet is communicated “to keep the TCP-variant connection active.” Thus, we have no issue with interpreting the words “no packet” in isolation to mean “no packet of any kind,” but we cannot ignore the phrase “to keep the TCP-variant connection active.” Thus, the phrase “no packet of any kind” is still qualified by other language in the claim.

Second, Patent Owner argues that the term “communicated” means “sent or received by the system.” Prelim. Resp. 14–23. According to Patent Owner, “since the claims refer to the connection in which the system is a part, ‘communicated’ refers to packets being sent or received by that system.” Prelim. Resp. 14. Stated differently, Patent Owner argues that “‘communicated’ should not be limited to *only* [‘receiving,’ ‘sending,’ or ‘receiving *and* sending.’ It is broader. Encompassing sending *or* receiving is consistent with the claim’s scope and claim differentiation.” Prelim. Resp. 17.

IPR2021-00628

Patent 10,075,564 B1

As explained below, we determine on this record that Petitioner has shown that the asserted prior art teaches “communicated” under Patent Owner’s proposed construction. Therefore, for purposes of this Decision, we apply Patent Owner’s construction of “communicated,” and we need not resolve any disputes between the parties over the scope of this term. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (“[W]e need only construe terms ‘that are in controversy, and only to the extent necessary to resolve the controversy’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

Third, Patent Owner argues that the phrase “to keep the TCP-variant connection active” means “to keep the TCP-variant connection active with or without a condition and via any mechanism.” Prelim. Resp. 9, 23–25 (emphasis omitted). As explained below, we find that the asserted prior art teaches, in at least one scenario, an absence of packets that are communicated to keep the connection active, and, therefore, we apply Patent Owner’s construction of the phrase “to keep the TCP-variant connection active” and need not resolve any disputes regarding the scope of this phrase.

2. *Remaining Terms*

For purposes of this Decision, we need not construe expressly any other claim terms. *See Nidec*, 868 F.3d at 1017.

D. Level of Ordinary Skill in the Art

Petitioner argues that a person having ordinary skill in the art “would have had an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking.” Pet. 5 (citing Ex. 1002

IPR2021-00628

Patent 10,075,564 B1

¶¶ 16–18). Petitioner further argues that “[m]ore education can supplement practical experience and vice versa.” Pet. 5 (citing Ex. 1002 ¶¶ 16–18).

Patent Owner substantially agrees with Petitioner’s proposal. Prelim. Resp. 5.

Accordingly, for purposes of this Decision, we adopt Petitioner’s proposed formulation of the level of ordinary skill in the art, except that we delete the qualifier “at least” to eliminate vagueness as to the amount of practical experience. The qualifier expands the range indefinitely without an upper bound, and thus precludes a meaningful indication of the level of ordinary skill in the art.

*E. Alleged Obviousness over Wookey and Eggert
(Claim 23)*

Petitioner asserts that claim 23 of the ’564 patent is unpatentable under 35 U.S.C. § 103 as obvious over the combined teachings of Wookey and Eggert. Pet. 12–74. Patent Owner opposes. Prelim. Resp. 25–57.

1. Wookey

Wookey “generally relates to providing access to computing environments” and, as relevant to this Decision, discloses an environment in which a client machine can access computing resources provided by various remote machines. Ex. 1005 ¶¶ 2, 135. Additional pertinent details are discussed below in the analysis of the parties’ contentions.

2. Eggert

Eggert “specifies a new TCP option - the Abort Timeout Option - that allows conforming hosts to negotiate per-connection abort timeouts. This allows mobile hosts to maintain TCP connections across disconnected periods that are longer than their system’s default abort timeout.” Ex. 1006, 3.

IPR2021-00628

Patent 10,075,564 B1

3. *Claim 23*

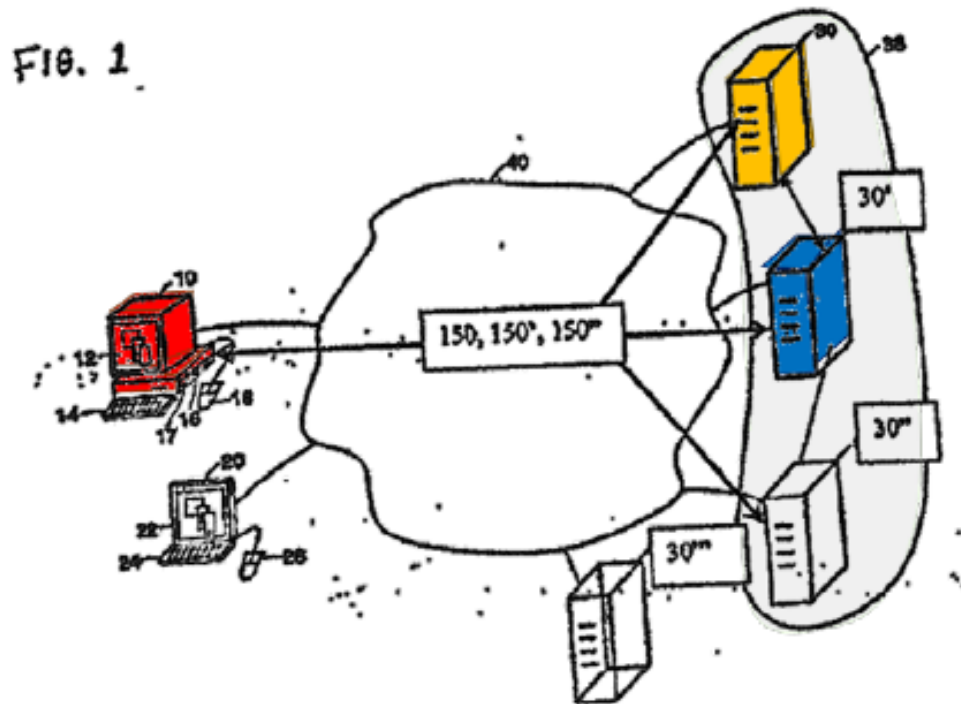
Claim 23 depends from disclaimed dependent claim 22, which in turn depends from disclaimed independent claim 16. Claim 16 is directed to “[a] non-transitory computer readable medium, comprising: code for being communicated to a remote client node.” Claims 16, 22, and 23 recite various operations that the client node performs based on the code involving communications with “a server node” and “another server node.” One set of operations involves “a transmission control protocol (TCP)-variant” packet and connection.

Petitioner relies on Wookey to teach communications between a client node and remote servers but notes that, although Wookey discloses using various transport protocols, “*Wookey* does not expressly provide details regarding establishing and maintaining the second connection or the types of negotiable parameters used to establish the second connection.” Pet. 25 (citing Ex. 1002 ¶ 130). Petitioner relies on Eggert in combination with Wookey for the subject matter pertaining to the TCP-variant protocol. Pet. 25–60.

Petitioner provides the following colored version of Wookey’s Figure 1 to show various components recited in claims 16, 22, and 23:

IPR2021-00628

Patent 10,075,564 B1



Pet. 13. Wookey's Figure 1, reproduced above with color added by Petitioner, "is a block diagram of one embodiment of an environment in which a client machine accesses a computing resource provided by a remote machine." Ex. 1005 ¶ 20. In the version of Wookey's Figure 1 reproduced above, Petitioner colors client machine 10 red, remote machine 30 orange, and remote machine 30' blue, and maps these, respectively, to "a client node," "another server node," and "a server node" recited in claims 16, 22, and 23. Pet. 13, 17, 57, 63. Petitioner argues that Wookey's client machine 10 includes main processor 102 ("one or more processors"), which is in communication with main memory unit 104 and cache memory 140, both of which are "non-transitory memory" according to Petitioner. Pet. 22 (citing Ex. 1005 ¶¶ 160–165, Figs. 1A, 1B; Ex. 1002 ¶¶ 117–119).

Petitioner argues that, in Wookey, client machine 10 executes web browser application 280 to transmit a request to remote machine 30, which responds to client machine 10 with HTML page 288 having encoded

IPR2021-00628

Patent 10,075,564 B1

Uniform Resource Locators (URLs) for resources that client machine 10 can access. Pet. 17–19 (citing 1005 ¶¶ 21, 24, 135–136, 154, 171, 192–193, 196, 207, 213, 215–216, 245, Figs. 1, 2C; Ex. 1002 ¶¶ 110–114). Wookey explains that “remote machine [30] prepares and transmits to the client machine 10 an HTML page 288 that includes a Resource Neighborhood window 258 in which appears graphical icons 257, 257' representing resources to which the client machine 10 has access.” Ex. 1005 ¶ 196. Petitioner argues that, “when a user clicks an icon 257, 257' corresponding to a resource from the displayed HTML page 288, client machine 10 uses the encoded URL associated with the clicked icon to initiate the process for establishing a second connection (e.g., connection 394 in Figure 3D below) with a second remote machine 30'.” Pet. 19–20 (emphasis omitted) (citing Ex. 1005 ¶¶ 26, 216; Ex. 1002 ¶ 114). For example, Wookey discloses that “[a] user of the client machine 10 can access a resource by clicking an icon 257, 257' displayed in the Resource Neighborhood web page,” after which “client machine 10 establishes a connection (arrow 394) with the remote machine 30' identified as hosting the requested resource and exchanges information regarding access to the desired resource.” Ex. 1005 ¶ 216.

Petitioner, therefore, contends that Wookey teaches communication between a client node and both “a server node” and “another server node,” as in claims 16, 22, and 23. Having set forth Petitioner’s contentions as to how Wookey teaches the overall client-to-server and client-to-“another server” communication scheme of these claims, we turn to Petitioner’s contentions as to the “code,” “TCP-variant,” “idle time period,” and “TCP” recitations of these claim.

IPR2021-00628

Patent 10,075,564 B1

a) “Code”

As noted above, claim 16, from which challenged claim 23 ultimately depends, is directed to “[a] non-transitory computer readable medium, comprising: code for being communicated to a remote client node.”

Claim 22, from which challenged claim 23 depends, recites “[a] system including the non-transitory computer readable medium of claim 16, wherein the system: includes another server node separate from the server node” and “is configured to communicate the code to the client node.”

Petitioner argues that Wookey teaches the recited “code” in two ways. First, Petitioner contends that Wookey’s encoded URLs are code, citing Wookey’s disclosure that the encoded URLs specify resource location, launch command for the resource, and information necessary for the client to connect to the remote machine. Pet. 19–20 (citing Ex. 1005 ¶¶ 26, 216; Ex. 1002 ¶¶ 114–115). According to Petitioner, therefore, the URLs are code because they “are instructions that, when invoked, command client machine 10 to access resources from one or more locations (e.g., one or more remote machines).” Pet. 20 (emphasis omitted) (citing Ex. 1002 ¶ 115).

Second, Petitioner argues that “HTML page 288 (including the encoded URLs) also discloses the claimed ‘code’ because an HTML page was known to comprise HTML code used to define, e.g., what is displayed on the screen of the client node.” Pet. 21 (citing Ex. 1002 ¶ 116; Ex. 1020 ¶¶ 22, 24, 26, 79).

Patent Owner argues that Wookey’s URL is not code but, rather, is an address. Prelim. Resp. 29–30 (citing Ex. 1005 ¶ 193). Patent Owner, however, does not contest Petitioner’s contentions that Wookey’s HTML is code. On this record, we are sufficiently persuaded by Petitioner’s

IPR2021-00628

Patent 10,075,564 B1

explanation that HTML is code because it defines what is displayed on a screen. *See* Ex. 1002 ¶ 116 (“[A]n HTML page was known as a set of HTML code used to define, e.g., what is displayed on the screen of the client node.”). Therefore, for purposes of this Decision, we need not rely on Petitioner’s contention that Wookey’s URL is “code.”

We also are persuaded that, in Wookey, remote machine 30 (“another server node”) sends HTML page 288 (“code”) to client machine 10 (“remote client node”). *See* Pet. 63. As noted above, Wookey explains that “remote machine [30] prepares and transmits to the client machine 10 an HTML page 288.” Ex. 1005 ¶ 196. Thus, on this record, we are persuaded that Wookey teaches a system that “includes another server node separate from the server node” and “is configured to communicate the code to the client node,” as recited in claim 22.

b) “TCP-Variant” Communication

Claim 16 recites that the code “results in the client node operating to” perform three operations relating to establishing a TCP-variant connection. Petitioner cites Wookey’s disclosure that the connection between client machine 10 and remote machine 30’ (“server node”) can be over various protocols, including the Virtual Network Computing (VNC) protocol, which can run over various protocols including TCP/IP, IPX/SPX, and NetBUI. Pet. 23–24 (citing Ex. 1005 ¶¶ 155, 215–216, 225; Ex. 1002 ¶¶ 122–123). In particular, Wookey discloses that the connection 394, which is between client machine 10 and remote machine 30’, can be over a number of protocols, including VNC, and that VNC can “run over industry standard transport protocols, such as TCP/IP, IPX/SPX, NetBEUI.” Ex. 1005 ¶¶ 216, 225. Petitioner argues that “*Wookey* does not expressly provide details regarding establishing and maintaining the second connection or the types of

IPR2021-00628

Patent 10,075,564 B1

negotiable parameters used to establish the second connection,” but Petitioner argues that Wookey does identify certain desired characteristics of the second connection, including “reducing unintentional termination of sessions due to an imperfect connection, detecting and handling disconnections like when a mobile device enters an elevator, and ability to specify inactive time prior to connection termination.” Pet. 24–25 (citing Ex. 1005 ¶¶ 581, 721, 738–739, 751, 1134–1136, 1153; Ex. 1002 ¶¶ 126–127, 130). Petitioner argues that Eggert discloses a TCP-variant protocol, which Petitioner relies on in combination with Wookey to teach the TCP-variant subject matter recited in claims 16, 22, and 23. Pet. 26–74.

Petitioner provides various rationales to support the proposed combination of Wookey and Eggert. Pet. 37–45. For example, Petitioner argues that a person of ordinary skill in the art would have “recognized the benefits of” combining Wookey and Eggert “because using the TCP variant protocol with an ATO option as described in Eggert would have provided client machine 10 with an improved, reliable connection to remote machine 30’ with the versatility of negotiated timeout parameters consistent with Wookey’s desired characteristics for the second connection.” Pet. 39 (emphasis omitted). Petitioner also argues that the proposed “combination would have involved the use of known technologies (e.g., aspects of similar protocols) and design concepts and processes to obtain the foreseeable result of a reliable connection between *Wookey*’s client machine 10 and remote machine 30’.” Pet. 41 (emphasis omitted) (citing Ex. 1002 ¶ 173).

Claim 16 recites three operations relating to establishing a TCP-variant connection, which we discuss below.

IPR2021-00628

Patent 10,075,564 B1

(1) *Identify idle information*

The first recited operation in claim 16 is “identify idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active.” Petitioner argues that Eggert’s disclosure of a new TCP option—the Abort Timeout Option (“ATO”)—teaches a “TCP-variant” protocol and that “a connection using *Eggert*’s ATO would be a TCP-variant connection.” Pet. 27 (citing Ex. 1006, Abstract, 3; Ex. 1002 ¶¶ 84–93, 135), 29 (citing Ex. 1002 ¶ 141). Petitioner argues that Eggert’s abort timeout value, which is exchanged in the ATO field in the three-way handshake of Figure 2, is “idle information” that “corresponds to an ‘idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active,’ as claimed.” Pet. 34 (citing Ex. 1002 ¶ 153). According to Petitioner, a person of ordinary skill in the art would have recognized that, in Eggert, an ACK (acknowledgement) packet is the packet that must be communicated to keep the connection active. Pet. 34 (citing Ex. 1006, 3; Ex. 1002 ¶ 154). Petitioner relies on (Pet. 34) the following disclosure from Eggert: “The TCP specification includes a ‘user timeout’ that defines the maximum amount of time that segments may remain unacknowledged before TCP will abort the connection.” Ex. 1006, 3 (citing RFC 793 (Ex. 1008 in the record)). Thus, Petitioner argues, “*Eggert*’s description of the abort timeout as the duration of time during which a node can wait to receive an ACK is a period ‘during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active.’” Pet. 34–35 (citing Ex. 1002 ¶ 155).

IPR2021-00628

Patent 10,075,564 B1

Petitioner also focuses on what happens in the Wookey/Eggert combination when there is a disconnection between client machine 10 and remote machine 30. *See* Pet. 45–48. This is sometimes referred to as the “disconnection hypothetical.” Specifically, Petitioner argues that “when an interruption in communication occurs such as a disconnection in the network service (as contemplated by *Wookey* and *Eggert*) or in a physical network medium (as described in the ’564 patent) (EX1001, 2:4–8, 2:23–25, 12:61–67), packets from one node do not reach the other node in either direction.” Pet. 45 (citing Ex. 1002 ¶ 177). According to Petitioner, “[i]n this situation contemplated by the ’564 patent, no packets are communicated between the nodes to keep the connection active because they cannot get through the interruption or the physical medium’s disconnection.” Pet. 45–46 (citing Ex. 1002 ¶ 177; Ex. 1001, 2:23–25, 12:61–67, Fig. 7).

Patent Owner argues that the combination of Wookey and Eggert fails to disclose the ITP subject matter. Prelim. Resp. 47–57. Patent Owner argues that ACK packets are not “the only packets keeping *Eggert*’s connection active” and cites “retransmitted-packets” as other packets that are “intended to keep the connection active.” Prelim Resp. 53. Patent Owner argues Petitioner’s expert admitted in a different proceeding that “retransmitted-packets *are* sent to keep the connection active.” Prelim. Resp. 53.⁷ Patent Owner also argues that Eggert does not disclose Petitioner’s disconnection hypothetical. Prelim. Resp. 49–52.

⁷ As support, Patent Owner directs us to exhibits filed in a different proceeding. However, “[a]ll evidence must be filed in the form of an exhibit.” 37 C.F.R. § 42.63(a). If a party wishes us to consider papers or exhibits from a different proceeding, the party must file them as an exhibit in this proceeding.

For the reasons explained below and based on the current record, we are sufficiently persuaded that Eggert’s abort timeout value in the ATO field represents a period where, if no packet is communicated to keep the TCP-variant connection active, the connection will abort. Our determination is based on what happens in Eggert—along with the combination of Eggert and Wookey—if there is a disconnection after an ATO period is negotiated in the three-way handshake.

Based on the current record, Eggert’s TCP-variant protocol uses the three-way handshake to establish a connection between two nodes on a network. Ex. 1006, 5; Ex. 1002 ¶ 142. The three-way handshake involves the exchange of synchronize (SYN) and ACK messages between the nodes to create a connection—i.e., SYN, SYN-ACK, and ACK messages in a standard TCP implementation. Ex. 1002 ¶ 142; Ex. 1008, 31–32, 34–37. During the three-way handshake, the client node and the server node may agree upon an abort timeout (ATO) period that is longer than the user timeout period set in the TCP specification, which “allows mobile hosts to maintain TCP connections across disconnected periods that are longer than their system’s default abort timeout.” Ex. 1006, 3. As part of that communication, the client node will receive a packet with the ATO parameter field and will then identify the metadata in that field establishing the ATO period. Ex. 1006, 5–7. That ATO period negotiated in Eggert sets the time period that must pass without an ACK before the connection is terminated. *See* Ex. 1002 ¶¶ 85–87, 154–155.

Based on the current record, a disconnection—such a disconnection in the network service or a physical network medium—between a client node and a server node using a Wookey/Eggert connection would cause the TCP-variant connection to abort the connection. Specifically, because of the

IPR2021-00628

Patent 10,075,564 B1

disconnection, no packets, including ACK packets, can get through the interruption. *See* Ex. 1002 ¶ 177. For example, if a packet is sent by the client node after the disconnection happens (*see* Pet. 45–47), the server node will not receive the packet and cannot send an ACK packet. Ex. 1002 ¶ 177. Once the original packet is sent and the ATO timer started, no ACK will be sent or received due to the disconnection, and, as a result, once the ATO timer has expired, the connection will be aborted. *See* Ex. 1002 ¶ 177; Ex. 1006, 3; Ex. 1008, 81. In other words, based on the current record, during the disconnection hypothetical, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active.

The retransmission timer cited by Patent Owner is a separate timer that governs retransmission of data if an ACK is not received within a particular timeout interval. *See* Prelim. Resp. 55 (citing Ex. 1008, 8, 14⁸). RFC 793—which is the Transmission Control Protocol Specification published by the DARPA internet program—explains that, “[w]hen the TCP transmits a segment containing data, it puts a copy on a retransmission queue and starts a timer; when the acknowledgment for that data is received, the segment is deleted from the queue.” Ex. 1008, 14. However, “[i]f the acknowledgment is not received before the timer runs out, the segment is retransmitted.” Ex. 1008, 14. There is no discussion of disconnecting the connection if a retransmission message is not sent. *Id.* Based on the current record, RFC 793 and Eggert do not disclose, and Patent Owner does not argue, that retransmitting data resets the original timeout timer that started upon sending the data the first time. Thus, despite retransmitting the data,

⁸ Patent Owner mistakenly cites Exhibit 1006 (Eggert), but the material quoted by Patent Owner is in Exhibit 1008 (RFC 793).

the connection will terminate on the expiration of the ATO timer that started when the first data were transmitted.

Patent Owner argues that the claims do not recite “any ‘mechanism’ to maintain the connection, such as via the ITP or an idle period timer, *e.g.* using a received ACK,” and, therefore, that the claims do not exclude retransmission packets from packets that keep the connection active. Prelim. Resp. 56. On this record, however, we do not agree that retransmission packets are packets that are sent to keep the connection active. If the purpose of retransmitting the data is to keep the connection active, then the device would be programmed to reset the ATO timer upon retransmission to avoid termination of the connection. The fact that this is not done supports a determination, based on the current record, that the purpose of these packets is not to keep the connection active. Instead, the purpose of the retransmission message is an attempt to retransmit data that was not properly received and acknowledged. For all of these reasons, based on the current record, we determine that retransmission packets in Eggert are not communicated “to keep the TCP-variant connection active.”

Accordingly, based on the current record and for the reasons given above, we are sufficiently persuaded by Petitioner’s contention that, under at least certain operating conditions, Eggert teaches the recited ITP subject matter, which is sufficient to show obviousness. *See Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016) (“[C]ombinations of prior art that sometimes meet the claim elements are sufficient to show obviousness.”); *Hewlett–Packard Co. v. Mustek Sys., Inc.*, 340 F.3d 1314, 1326 (Fed. Cir. 2003) (“[A] prior art product that sometimes, but not always, embodies a claimed method nonetheless teaches that aspect of the invention.”).

(2) Generate a TCP-variant packet

The second recited operation in claim 16 is “generate a TCP-variant packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information.” Petitioner argues that Eggert discloses generating packets having the ATO value, thereby teaching this subject matter. Pet. 48–54 (citing Ex. 1006, 3–7, 9, Figs. 1, 2; Ex. 1002 ¶¶ 181–193). For example, Eggert discloses that the ATO value is included in any segment (packet) that has “a SYN flag, i.e., either the initial SYN or the SYN-ACK.” Ex. 1006, 5. Eggert’s Figure 2 illustrates “Allowed TCP Abort Timeout Option (ATO) Exchanges” and shows the ATO included in SYN and SYN/ACK packets. Ex. 1006, 7.

(3) Send the TCP-variant packet

The third recited operation in claim 16 is “send, from the client node to a server node, the TCP-variant packet to provide the metadata for the idle time period to the server node, for use by the server node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.” Petitioner contends that Eggert’s disclosure of exchanging packets with the ATO value between nodes and accepting or shortening the abort timeout attribute based on the exchanged ATO value, in combination with Wookey’s client-server architecture, teaches this subject matter. Pet. 54–60 (citing Ex. 1006, 3, 5–7, 9, 11, Fig. 2; Ex. 1002 ¶¶ 194–213). For example, Eggert discloses that, “[u]pon receipt of a segment with the Abort Timeout Option, the receiving host decides whether to accept, shorten, or reject its peer’s proposed abort timeout.” Ex. 1006, 5.

c) Code for Establishing TCP and TCP-Variant Connections

As discussed above in § II.E.3.a, we are sufficiently persuaded, on this record, that Wookey teaches a system that “includes another server node

IPR2021-00628

Patent 10,075,564 B1

separate from the server node” and “is configured to communicate the code to the client node,” as recited in claim 22. Claim 22 recites that the system “is further configured to perform a 3-way TCP handshake for establishing, with the client node, a TCP connection that is different than the TCP-variant connection for permitting the client node to fetch the code in addition to other data from the system via a hypertext transfer protocol (HTTP).”

Petitioner cites Wookey’s disclosure of various protocols that can be used for communications, including TCP/IP. Pet. 23–24 (citing Ex. 1005 ¶¶ 155, 159, 215–216, 225). Wookey explains that “communications link 150 may use a transport layer protocol such as TCP/IP.” Ex. 1005 ¶ 155.

Communications link 150 is between client machine 10 and remote machine 30. Ex. 1005 ¶ 154 (“[T]he client machine 10 communicates with the remote machine 30 in the machine farm 38 over a communications link 150.”).

Petitioner argues that “a three-way handshake method is necessarily performed to establish a TCP connection” because the TCP specification (RFC 793) requires a three-way handshake to establish a TCP connection. Pet. 65–66 (citing Ex. 1002 ¶¶ 229–230; Ex. 1008 (RFC 793), 31–32, 34–37); *see* Ex. 1008, 34 (“The ‘three-way handshake’ is the procedure used to establish a connection.”), 32 (“A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN’s.”). Petitioner further argues that “[t]he TCP connection established between client machine 10 and remote machine 30 permits the client machine to receive the code via an HTTP-based application (e.g., the web server and web browser application 280).” Pet. 68 (emphasis omitted) (citing Ex. 1002 ¶¶ 237–238).

Claim 23 recites,

IPR2021-00628

Patent 10,075,564 B1

The system of claim 22, wherein the code, when used by the client node, results in the client node operating such that the TCP-variant connection is established between the client node and the server node instead of the TCP connection in order to permit communication, between the client node and the server node, of the timeout attribute, where the timeout attribute is not communicated when establishing the TCP connection between the system and the client node, but is communicated when establishing the TCP-variant connection between the client and the server node.

Petitioner refers to its contentions for claim 16 regarding the TCP-variant connection that is set up between client machine 10 and remote machine 30', which includes Eggert's ATO exchange, and Petitioner argues that the ATO value is not communicated during setup of the TCP connection. Pet. 71–74 (citing Ex. 1007, 18:56–21:35, 24:40–47; Ex. 1002 ¶¶ 244–253).

Patent Owner disputes Petitioner's contentions for certain of the "TCP" and "TCP-variant" subject matter recited in the challenged claims. Prelim. Resp. 25–27. Patent Owner argues that these claims recite "two separate, but related, requirements": (1) "that the client be able to fetch the 'code' via HTTP using a TCP connection different than the TCP-variant connection ('Claim Requirement #1['])" and (2) "that use of the fetched code by the client node results in the TCP-variant connection being established instead of a TCP connection ('Claim Requirement #2['])." Prelim. Resp. 26 (footnote omitted). Patent Owner argues that Petitioner's contentions fail because they rely on different embodiments in Wookey to show these two claim requirements—Figure 3D for "Claim Requirement #1" and Figure 25, as discussed in paragraph 732, for "Claim Requirement #2." Prelim. Resp. 27–29. According to Patent Owner, there is no "nexus

IPR2021-00628

Patent 10,075,564 B1

between *Wookey*'s First (Fig. 3D) and Second Embodiments (Fig. 25)."

Prelim. Resp. 30.

On this record, we disagree with Patent Owner's arguments. Petitioner cites paragraph 732 of *Wookey* to show that client machine 10 can communicate with remote machine 30 using a protocol that is different from the one used for communications with remote machine 30'. *See* Pet. 24 (citing Ex. 1005 ¶ 732). As an example of this, *Wookey* explains that "client machine 10 may make a request to the remote machine 30 using the IPX protocol and request the address of the remote machine 30' as a TCP/IP protocol address." Ex. 1005 ¶ 732. But this is not the only disclosure in *Wookey* on which Petitioner relies to show the use of different protocols for different connections. For example, Petitioner cites paragraph 155 in *Wookey*, which, as noted above, states that communications link 150 between client machine 10 and remote machine 30 "may use a transport layer protocol such as TCP/IP." *See* Pet. 23–24 (citing, *inter alia*, Ex. 1005 ¶ 155). Petitioner also cites the use of VNC for the connection between client machine 10 and remote machine 30' and notes that "VNC can run over various 'industry standard transport protocols, such as TCP/IP, IPX/SPX, NetBEUI.'" Pet. 23–24 (quoting Ex. 1005 ¶ 225). On this record, we are sufficiently persuaded by *Wookey*'s express disclosure of TCP/IP communications between client machine 10 and remote machine 30.

On this record, we also disagree with Patent Owner's arguments that the "code" identified by Petitioner does not "result[] in" the establishment of a TCP-variant connection between the client node and the server node instead of a TCP connection. *See* Prelim. Resp. 29–32. Rather, on this record, we are sufficiently persuaded by Petitioner's contentions, summarized above, that *Wookey*'s HTML page 288, when used by client

IPR2021-00628

Patent 10,075,564 B1

machine 10, results in a connection being made to remote machine 30' ("server node"). *See* Pet. 71–72. For example, Wookey explains that "HTML page 288 . . . includes a Resource Neighborhood window 258 in which appears graphical icons 257, 257' representing resources to which the client machine 10 has access." Ex. 1005 ¶ 196. Wookey discloses that "[a] user of the client machine 10 can access a resource by clicking an icon 257, 257' displayed in the Resource Neighborhood web page," after which "client machine 10 establishes a connection (arrow 394) with the remote machine 30' identified as hosting the requested resource and exchanges information regarding access to the desired resource." Ex. 1005 ¶ 216. On this record, we also are persuaded by Petitioner's contentions, summarized above, that it would have been obvious to a person of ordinary skill in the art to use Eggert's teachings to implement a TCP-variant connection between client machine 10 and remote server 30' in Wookey. *See* Pet. 23–45.

d) Determination for Claim 23

On this record, we are sufficiently persuaded by Petitioner's contentions and evidence, summarized above, and we disagree with Patent Owner's arguments. In particular, we are persuaded that Petitioner has explained sufficiently how the proposed combination of Wookey and Eggert teaches the subject matter recited in claim 23 and that a person of ordinary skill in the art would have had reason to combine the teachings of Wookey and Eggert in the manner asserted. Having considered the parties' arguments, we determine that Petitioner has established a reasonable likelihood of prevailing in challenging claim 23 as obvious over the combined teachings of Wookey and Eggert.

IPR2021-00628

Patent 10,075,564 B1

III. CONCLUSION

For the foregoing reasons, we determine that the information presented in the Petition establishes that there is a reasonable likelihood that Petitioner would prevail in challenging claim 23 of the '564 patent, and we institute *inter partes* review of claim 23 on the sole ground raised in the Petition. See PTAB Rules of Practice for Instituting on All Challenged Patent Claims and All Grounds and Eliminating the Presumption at Institution Favoring Petitioner as to Testimonial Evidence, 85 Fed. Reg. 79,120, 79,129 (Dec. 9, 2020) (codified at 37 C.F.R. 42.108(a)); *see also* Prelim. Resp. 1 (“Jenam statutorily disclaimed claims 16 and 22. Only claim 23 remains.”). At this stage of the proceeding, we have not made a final determination with respect to the patentability of claim 23 or the construction of any claim term.

IV. ORDER

Accordingly, it is

ORDERED that pursuant to 35 U.S.C. § 314(a) and 37 C.F.R. § 42.4, an *inter partes* review is hereby instituted as to claim 23 of the '564 patent on all challenges raised in the Petition; and

FURTHER ORDERED that, pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial, which will commence on the entry date of this decision.

IPR2021-00628

Patent 10,075,564 B1

FOR PETITIONER:

Naveen Modi

Joseph Palys

Quadeer Ahmed

Jason Heidemann

PAUL HASTINGS LLP

naveenmodi@paulhastings.com

josephpalys@paulhastings.com

quadeerahmed@paulhastings.com

jasonheidemann@paulhastings.com

FOR PATENT OWNER:

Timothy Devlin

DEVLIN LAW FIRM LLC

td-ptab@devlinlawfirm.com

EXHIBIT 15

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

JENAM TECH, LLC,
Patent Owner.

IPR2021-00629
Patent 10,075,564 B1

Before DANIEL J. GALLIGAN, SCOTT B. HOWARD, and
JASON M. REPKO, *Administrative Patent Judges*.

GALLIGAN, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

IPR2021-00629

Patent 10,075,564 B1

I. INTRODUCTION

A. Background

Google LLC (“Petitioner”) filed a Petition requesting *inter partes* review of claims 24–29 of U.S. Patent No. 10,075,564 B1 (“the ’564 patent,” Ex. 1001). Paper 1 (“Pet.”). Jenam Tech, LLC (“Patent Owner”) filed a Preliminary Response, in which Patent Owner stated that it had “statutorily disclaimed claims 24, 28, and 29,” leaving only claims 25–27 for consideration. Paper 7 (“Prelim. Resp.”), 1.

With our authorization, Petitioner and Patent Owner filed, respectively, a Reply (Paper 9) and a Sur-reply (Paper 10) addressing discretionary denial issues.

Under 37 C.F.R. § 42.4(a), we have authority to determine whether to institute review. The standard for instituting an *inter partes* review is set forth in 35 U.S.C. § 314(a), which provides that an *inter partes* review may not be instituted unless the information presented in the Petition and the Preliminary Response shows “there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.”

For the reasons explained below, we institute an *inter partes* review as to challenged claims 25–27 on the sole ground raised in the Petition.

B. Related Matters

As required by 37 C.F.R. § 42.8(b)(2), the parties identify various related matters, including another petition for *inter partes* review challenging different claims of the ’564 patent. Pet. 1–2; Paper 5; *see also* IPR2021-00628, Paper 1 (challenging claims 16, 22, and 23 of the ’564 patent).

IPR2021-00629

Patent 10,075,564 B1

C. Real Parties in Interest

The parties identify themselves as the real parties in interest. Pet. 1; Paper 5.

D. The '564 Patent and Illustrative Claims

The '564 patent states that some transmission control protocol (TCP) implementations support a keep-alive option, but the '564 patent states that “[e]ach node supports or does not support keep-alive for a TCP connection based on each node’s requirements without consideration for the other node in the TCP connection.” Ex. 1001, 1:48–50, 1:58–61. According to the '564 patent, “[t]o date no mechanism to allow two TCP connection endpoints to cooperate in supporting the keep-alive option has been proposed or implemented.” Ex. 1001, 2:19–22. Thus, the '564 patent states that “there exists a need for methods, systems, and computer program products for sharing information for detecting an idle TCP connection.” Ex. 1001, 2:26–28.

Remaining challenged claims 25–27 depend from claim 24. All four of these claims are reproduced below.

24. A non-transitory computer readable medium, comprising:
code for use by a client node including one or more processors in communication with a non-transitory memory storing a network application that is configured to operate in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, where the code, when used by the client node, results in the client node utilizing the network application to operate in accordance with the non-TCP protocol to:
receive, from a server node, a non-TCP packet during a setup of a non-TCP connection;
identify metadata, that specifies a number of seconds or minutes, in an idle time period parameter field

IPR2021-00629

Patent 10,075,564 B1

in the non-TCP packet, for an idle time period, where, as a result of a detection of the idle time period, the non-TCP connection is subject to deactivation; and

determine, based on the metadata, a timeout attribute associated with the non-TCP connection;

wherein the code, when used by the client node, results in the client node utilizing the network application to operate in accordance with the TCP protocol to perform a three-way TCP handshake with another server node for setting up a TCP connection with the another server node that is different than the non-TCP connection.

25. A system including the another server node which includes the non-transitory computer readable medium of claim 24, wherein the system is configured to communicate the code to the client node utilizing the TCP connection, for setting up the non-TCP connection with the server node.

26. The system of claim 25, wherein the system is configured to communicate the code to the client node via the TCP connection utilizing a hypertext transfer protocol (HTTP).

27. The system of claim 26, wherein the code, when used by the client node, results in the client node operating such that the non-TCP connection is setup between the client node and the server node instead of the TCP connection in order to permit communication, between the client node and the server node, of the timeout attribute, where the timeout attribute is not communicated when setting up the TCP connection between the system and the client node, but is communicated when establishing the non-TCP connection between the client and the server node.

IPR2021-00629

Patent 10,075,564 B1

E. Asserted Ground of Unpatentability

Petitioner challenges the patentability of claims 25–27 of the ’564 patent on the following ground:

Claim(s) Challenged	35 U.S.C. §	Reference(s)/Basis
25–27	103	Wookey, ¹ Berg ²

II. ANALYSIS

*A. Discretionary Denial**1. 35 U.S.C. § 314(a)*

Patent Owner argues that we should exercise discretion to deny the Petition under 35 U.S.C. § 314(a) based on the factors set forth in *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 (PTAB Sept. 6, 2017) (designated precedential in relevant part). Prelim. Resp. 37–50. Patent Owner cites other petitions for *inter partes* review that have been filed challenging claims of Patent Owner’s patents and argues that, “[b]ecause [Petitioner’s] petitions challenge related patents that cover the same subject matter and include overlapping claim limitations, these follow-on petitions place an unfair burden on Patent Owner and should be denied.” Prelim. Resp. 36–37.

We do not agree with Patent Owner’s argument that this is the type of “follow-on” petition that should be denied under the Board’s precedential *General Plastic* decision. *General Plastic* sets forth a series of factors to be considered by the Board in evaluating whether to exercise discretion under § 314(a) to deny a petition that challenges a patent that was previously

¹ Wookey, US 2007/0171921 A1, published July 26, 2007 (Ex. 1005).

² Berg, US 6,674,713 B1, issued Jan. 6, 2004 (Ex. 1007).

IPR2021-00629

Patent 10,075,564 B1

challenged before the Board. *General Plastic*, Paper 19 at 15–19. As *General Plastic* states, “[m]ultiple, staggered petitions *challenging the same patent and same claims* raise the potential for abuse.” *Id.* at 17 (emphasis added). Here, Patent Owner cites numerous petitions against related patents but only one other petition challenging claims of the ’564 patent. Prelim. Resp. 36–37. The other petition challenging the ’564 patent (IPR2021-00628, Paper 1) was filed the same date as the present Petition and challenges different claims of the ’564 patent. Prelim. Resp. 36; Paper 4 (according filing date of March 15, 2021); IPR2021-00628, Paper 4 (according filing date of March 15, 2021). Thus, this is not a situation involving “[m]ultiple, staggered petitions challenging the same patent and same claims” contemplated by *General Plastic*. See *General Plastic*, Paper 19 at 17. Rather, this is a situation involving parallel petitions, which we address below in § II.A.3.

Patent Owner argues that “the Board has previously denied institution when related, follow-on petitions challenged claims similar in scope and content, even when the claims were in different patents.” Prelim. Resp. 42 (citing *Tietex Int’l, LTD., v. Precision Fabrics Group, Inc.*, IPR2015-01671, Paper 7, at 8 (PTAB Feb. 11, 2016)). According to Patent Owner, “[t]he Board should similarly exercise its discretion to deny this Petition. Otherwise [Patent Owner] would be forced to expend resources defending claims that are similar in scope and content to those of [U.S. Patent 9,923,995 B1 (“the ’995 patent”)] it is already defending.” Prelim. Resp. 42.

The decision denying institution in *Tietex*, which is not precedential, is readily distinguishable from the present Petition. In *Tietex*, the panel

IPR2021-00629

Patent 10,075,564 B1

denied institution on one ground³ where the petition presented “the same arguments that we[re] found to be unpersuasive in” a previous final written decision addressing claims of similar scope in a related patent. *Tietex*, Paper 7 at 8. By contrast, we have issued a final written decision in IPR2020-00845 holding all challenged claims of the ’995 patent unpatentable. *See Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845, Paper 33 (PTAB Aug. 27, 2021). Therefore, the reasoning in *Tietex* does not apply to the facts of the present case.

For the foregoing reasons, we do not deny the Petition based on *General Plastic* under § 314(a).

2. 35 U.S.C. § 325(d)

Patent Owner argues that the Petition should be denied under 35 U.S.C. § 325(d) because Berg was previously presented to the Office in IPR2020-00742 (“the ’742 IPR”) and “[t]he arguments relating to Berg in the instant Petition are substantially similar to those set forth in the ’742 IPR.” Prelim. Resp. 51–52 (citing *Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6, 7–8 (PTAB Feb. 13, 2020) (precedential)). We disagree with Patent Owner’s argument because the ’742 IPR involved a different patent (the ’995 patent), and our precedent states that, “[u]nder § 325(d), the art and arguments must have been previously presented to the Office during proceedings pertaining to *the challenged patent*.” *Advanced Bionics*, Paper 6 at 7 (emphasis added).

³ The *Tietex* decision issued before the Supreme Court’s holding that a decision to institute under 35 U.S.C. § 314 may not institute on fewer than all claims challenged in the petition. *See SAS Inst., Inc. v. Iancu*, 138 S. Ct. 1348, 1359–60 (2018).

IPR2021-00629

Patent 10,075,564 B1

Thus, we do not deny the Petition under § 325(d) based on the use of Berg in the '742 IPR.⁴

3. Multiple Petitions

Patent Owner argues that “the November 2019 Consolidated Trial Practice Guide states that petitioners that file multiple petitions against the same patent must submit” a ranking of the petitions and an explanation of the differences. Prelim. Resp. 50 (citing Consolidated Trial Practice Guide (Nov. 2019), 59–60). Patent Owner argues that Petitioner “failed to submit this information” and, therefore, that the Petition should be denied. Prelim. Resp. 50.

Petitioner, however, did submit this information as Paper 3 in this proceeding. Petitioner explains that it filed different petitions to address different claim sets—one set (claims 16, 22, and 23) reciting “TCP-variant” and the other set (claims 24–29) reciting “non-TCP.” Paper 3, 2. Petitioner argues that,

[g]iven the difference in the protocol used between claims 16, 22, and 23 and claims 24–29 of the '564 patent, as well as the numerous elements in each of the claims (including the dependent claims), Petitioner submits that challenging all claims of the '564 patent in a single petition is not feasible or administratively efficient.

Paper 3, 2.

Patent Owner responds that Petitioner “could have asserted both bases for challenging the claims of the '564 patent in a single Petition” but filed two petitions instead. Prelim. Resp. 50. According to Patent Owner,

⁴ The '742 IPR was terminated based on a settlement between the parties to that proceeding before a final written decision issued. IPR2020-00742, Paper 31.

IPR2021-00629

Patent 10,075,564 B1

[t]he effect of Google breaking up its challenge to the '564 patent into two separate Petitions is to have substantially more pages to devote to its two invalidity theories, which substantially increases the workload for the Board and Jenam in analyzing and responding to what could have been set forth more succinctly in a single petition.

Prelim. Resp. 51.

We find Petitioner's explanation for filing two petitions challenging claims of the '564 patent to be satisfactory. Each petition challenges different claims and presents only one ground against each claim. Paper 3, 1; Pet. 3 (challenging claims 24–29 based on the combination of Wookey and Berg); IPR2021-00628, Paper 1, 3 (challenging claims 16, 22, and 23 based on the combination of Wookey and Eggert⁵). Therefore, this is not a situation where a petitioner is using two petitions to lodge multiple parallel challenges to the same claims. Also, as Petitioner points out (Paper 3, 2), the two petitions challenge different claim sets that are directed to different protocols. Claims 24–29 recite “non-TCP connection” and “non-TCP packet,” whereas claims 16, 22, and 23 recite “TCP-variant connection” and “TCP-variant packet.” Petitioner relies on different references to account for these different protocols—Eggert for “TCP-variant” and Berg for “non-TCP.” Paper 3, 3–4. Thus, Petitioner's “two invalidity theories” (Prelim. Resp. 51) address different recited subject matter.

Furthermore, the present Petition challenges two independent claims and additional claims dependent from the independent claims. Independent claim 24 is directed to “[a] non-transitory computer readable medium,” and independent claim 28 is directed to “[a] computer-implemented method.”

⁵ L. Eggert, “TCP Abort Timeout Option,” Apr. 14, 2004 (IPR2021-00628, Ex. 1006).

IPR2021-00629

Patent 10,075,564 B1

Although Patent Owner has disclaimed claim 28 and dependent claim 29 (Prelim. Resp. 1), this happened after Petitioner filed the Petition.

According to Petitioner, Patent Owner has asserted claim 29 against Petitioner in related district court litigation. Paper 3, 2. Thus, the Petition may be longer than it needs to be only because Patent Owner asserted a claim only to later disclaim that claim and its base claim.

For the above reasons, we decline to exercise our discretion to deny the Petition based on Petitioner's filing two petitions challenging the '564 patent.

B. Patent Owner's Arthrex Argument

Patent Owner argues that, based on the Supreme Court's decision in *United States v. Arthrex, Inc.*, 141 S. Ct. 1970 (2021), "the Board does not have authority to decide issues of patentability presented in a petition for inter partes review" until procedures are put in place by which the Director can review decisions by Administrative Patent Judges. Prelim. Resp. 52–53. We disagree with Patent Owner's argument because the Court in *Arthrex* held that "Congress has vested the Director with the 'power and duties' of the PTO, § 3(a)(1)," and that "[t]he Director accordingly may review final PTAB decisions and, upon review, may issue decisions himself on behalf of the Board." *Arthrex*, 141 S. Ct. at 1987. Because the Director has the authority to review final decisions of the Board, we disagree with Patent Owner's suggestion that the absence of specific review procedures results in a Constitutional infirmity.

Furthermore, Patent Owner's argument is moot because the Office has implemented a Director review procedure per *Arthrex*. See <https://www.uspto.gov/patents/patent-trial-and-appeal->

IPR2021-00629

Patent 10,075,564 B1

board/procedures/uspto-implementation-interim-director-review.

Specifically,

Parties may request Director review of a final written decision in an inter parties review or a post-grant review by concurrently (1) entering a Request for Rehearing by the Director into PTAB E2E and (2) submitting a notification of the Request for Rehearing by the Director to the Office by email to Director_PTABDecision_Review@uspto.gov, copying counsel for all parties by email.

Id.

Thus, we decline to deny institution based on the Supreme Court's decision in *Arthrex*.

C. Principles of Law

A patent claim is unpatentable under 35 U.S.C. § 103 if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) any secondary considerations, if in evidence.⁶ *Graham v. John Deere Co. of Kan. City*, 383 U.S. 1, 17–18 (1966).

⁶ Patent Owner does not present any objective evidence of nonobviousness (i.e., secondary considerations) as to any of the challenged claims.

D. Claim Construction

We interpret claim terms using “the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. § 42.100(b) (2019).

For purposes of this Decision, we need not construe expressly any claim terms. *See, e.g., Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (“[W]e need only construe terms ‘that are in controversy, and only to the extent necessary to resolve the controversy’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

E. Level of Ordinary Skill in the Art

Petitioner argues that a person having ordinary skill in the art “would have an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking.” Pet. 4–5 (citing Ex. 1002 ¶¶ 16–18). Petitioner further argues that “[m]ore education can supplement practical experience and vice versa.” Pet. 5 (citing Ex. 1002 ¶¶ 16–18).

Patent Owner substantially agrees with Petitioner’s proposal. Prelim. Resp. 5.

Accordingly, for purposes of this Decision, we adopt Petitioner’s proposed formulation of the level of ordinary skill in the art, except that we delete the qualifier “at least” to eliminate vagueness as to the amount of practical experience. The qualifier expands the range indefinitely without an

IPR2021-00629

Patent 10,075,564 B1

upper bound, and thus precludes a meaningful indication of the level of ordinary skill in the art.

*F. Alleged Obviousness over Wookey and Berg
(Claims 25–27)*

Petitioner asserts that claims 25–27 of the '564 patent are unpatentable under 35 U.S.C. § 103 as obvious over the combined teachings of Wookey and Berg. Pet. 11–63. Patent Owner opposes. Prelim. Resp. 6–34.

1. Wookey

Wookey “generally relates to providing access to computing environments” and, as relevant to this Decision, discloses an environment in which a client machine can access computing resources provided by various remote machines. Ex. 1005 ¶¶ 2, 135. Additional pertinent details are discussed below in the analysis of the parties’ contentions.

2. Berg

Berg relates to packet network communications and discloses negotiating various parameters when using Reliable User Datagram Protocol (RUDP). Ex. 1007, 17:9–17, 18:56–19:40. Berg discloses that a synchronization (SYN) segment “is used to establish a connection and synchronize sequence numbers between two hosts.” Ex. 1007, 18:57–58. The SYN segment contains various negotiable parameters, including a “Null Segment Timeout Value,” which specifies “[t]he timeout value for sending a null segment if a data segment has not been sent,” and a “Transfer State Timeout Value,” which specifies “the amount of time the state information will be saved for a connection after an auto reset occurs.” Ex. 1007, 18:58–60, 20:37–38, 20:43–45. Berg explains that a client sends a null segment to

IPR2021-00629

Patent 10,075,564 B1

the server if a certain period of time (the null segment timeout) elapses with no data segment having been sent. Ex. 1007, 23:46–49.

3. *Claims 25–27*

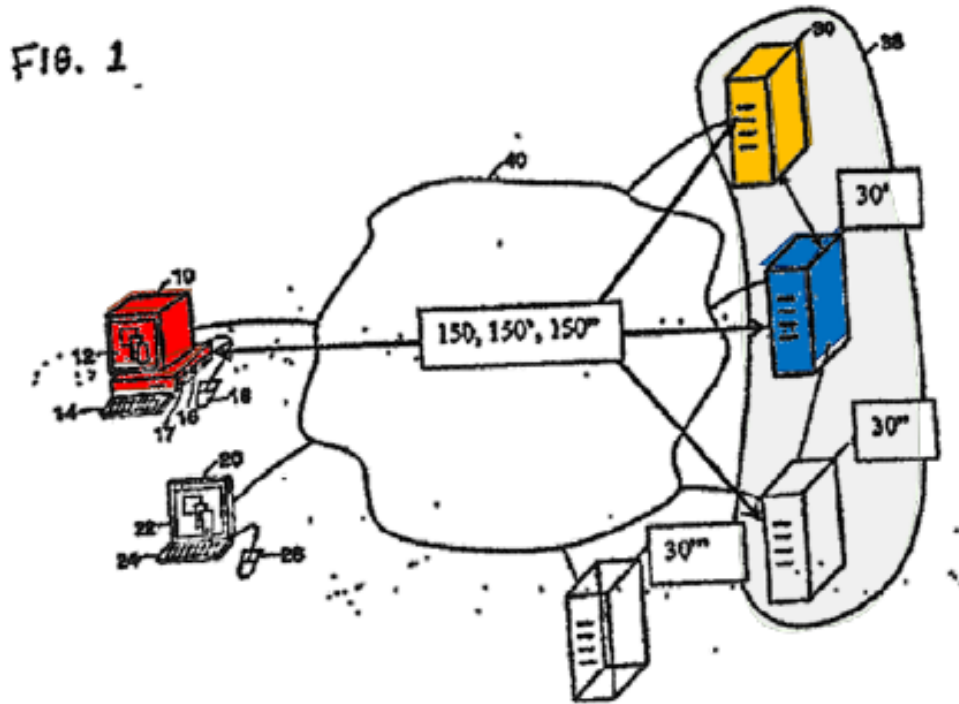
Claim 25 depends from disclaimed independent claim 24, and, therefore, claim 25 incorporates all of the subject matter recited in claim 24. Claim 26 depends from claim 25, and claim 27 depends from claim 26. Claim 24 is directed to “[a] non-transitory computer readable medium, comprising: code for use by a client node.” Claim 24 then recites various features of the client node and operations that the client node performs based on the code involving communications with “a server node” and “another server node.” One set of operations involves using “a non-transmission control protocol (TCP) protocol.”

Petitioner relies on Wookey for much of the subject matter recited in claim 25 but notes that, although Wookey discloses using non-TCP protocols, “Wookey does not expressly provide details regarding establishing and maintaining the second connection or the types of negotiable parameters used to establish the second connection.” Pet. 22. Petitioner relies on Berg in combination with Wookey for the subject matter pertaining to the non-TCP protocol. Pet. 23–50.

Petitioner provides the following colored version of Wookey’s Figure 1 to show various components recited in claim 24:

IPR2021-00629

Patent 10,075,564 B1



Pet. 12. Wookey's Figure 1, reproduced above with color added by Petitioner, "is a block diagram of one embodiment of an environment in which a client machine accesses a computing resource provided by a remote machine." Ex. 1005 ¶ 20. In the version of Wookey's Figure 1 reproduced above, Petitioner colors client machine 10 red, remote machine 30 orange, and remote machine 30' blue, and maps these, respectively, to "a client node," "another server node," and "a server node" recited in claim 24. Pet. 12, 15, 38, 51. Petitioner argues that Wookey's client machine 10 "includes a CPU 102 ('one or more processors') in communication with a main memory unit 104 and cache memory 140 (individually or collectively, 'non-transitory memory') storing a web browser application 280 ('network application')." Pet. 33–34 (citing Ex. 1005 ¶¶ 21, 160, 162–165, Figs. 1A, 1B; Ex. 1002 ¶¶ 155–156).

Petitioner argues that, in Wookey, client machine 10 executes web browser application 280 (the recited "network application") to transmit a

IPR2021-00629

Patent 10,075,564 B1

request to remote machine 30, which responds to client machine 10 with HTML page 288 having encoded Uniform Resource Locators (URLs) for resources that client machine 10 can access. Pet. 15–18 (citing 1005 ¶¶ 20, 24, 192–193, 196, 207, 213, 216, 754, 757, Fig. 2C; Ex. 1002 ¶¶ 115–118). Wookey explains that “remote machine [30] prepares and transmits to the client machine 10 an HTML page 288 that includes a Resource Neighborhood window 258 in which appears graphical icons 257, 257' representing resources to which the client machine 10 has access.” Ex. 1005 ¶ 196. Petitioner argues that, “when a user clicks an icon 257, 257' from the displayed HTML page 288, client machine 10 utilizes the web browser 280 to use the encoded URL associated with the clicked icon to initiate the process for establishing a second connection (e.g., connection 394 in Figure 3D below) with remote machine 30'.” Pet. 18 (emphasis omitted) (citing Ex. 1005 ¶ 216). For example, Wookey discloses that “[a] user of the client machine 10 can access a resource by clicking an icon 257, 257' displayed in the Resource Neighborhood web page,” after which “client machine 10 establishes a connection (arrow 394) with the remote machine 30' identified as hosting the requested resource and exchanges information regarding access to the desired resource.” Ex. 1005 ¶ 216.

Petitioner also argues that Wookey teaches that client machine 10 may access a resource on remote machine 30 (“another server node”). Pet. 51–52 (citing Ex. 1005 ¶¶ 6, 207, 213, 216, 625, 627, 673, 707; Ex. 1002 ¶¶ 202–203). Wookey discloses that, “[i]n other embodiments, the first remote machine 30 is hosting the selected one of the available resources.” Ex. 1005 ¶ 627. Wookey explains that, if remote machine 30 is hosting the resource, then the user can access the resource “over the existing connection” or “over a new connection” with remote machine 30. Ex. 1005 ¶ 627.

Petitioner, therefore, contends that Wookey teaches communication between a client node and both “a server node” and “another server node,” as in claim 24. Having set forth Petitioner’s contentions as to how Wookey teaches the overall client-to-server and client-to-“another server” communication scheme of claim 24, we turn to Petitioner’s contentions as to the “code,” “TCP,” and “non-TCP” recitations of claim 24.

a) “Code”

As noted above, claim 24 is directed to “[a] non-transitory computer readable medium, comprising: code for use by a client node.”

Petitioner argues that Wookey teaches “code for use by a client node” in two ways. First, Petitioner contends that Wookey’s encoded URLs are code, citing Wookey’s disclosure that the encoded URLs specify resource location, launch command for the resource, and information necessary for the client to connect to the remote machine. Pet. 17–18 (citing Ex. 1005 ¶ 216; Ex. 1002 ¶¶ 117–119). According to Petitioner, therefore, the URLs are code because they “are instructions that, when used by client machine 10, command client machine 10 to access resources from one or more remote machines.” Pet. 18 (emphasis omitted).

Second, Petitioner argues that “HTML page 288 (including the encoded URLs and icons) also discloses the claimed ‘code for use by a client node’ because an HTML page contains HTML code used to define, e.g., what is displayed on the screen of client machine 10.” Pet. 19 (citing Ex. 1002 ¶ 124; Ex. 1012 ¶¶ 22, 24, 26, 79).

Patent Owner argues that Wookey’s URL is not code but, rather, is an address. Prelim. Resp. 13–14 (citing Ex. 1005 ¶ 193). Patent Owner, however, does not contest Petitioner’s contentions that Wookey’s HTML is code. On this record, we are sufficiently persuaded by Petitioner’s

IPR2021-00629

Patent 10,075,564 B1

explanation that HTML is code because it defines what is displayed on a screen. *See* Ex. 1002 ¶ 124 (“[A]n HTML page includes HTML code used to define what is displayed on the screen of the client node.”). Therefore, for purposes of this Decision, we need not rely on Petitioner’s contention that Wookey’s URL is “code.”

b) “TCP” and “Non-TCP” Communications

Claim 24 recites that the code results in the client node operating according to a “non-TCP protocol” to communicate with “a server node” and according to “the TCP protocol” to communicate with “another server node.”

As for the recitations of claim 24 involving TCP and non-TCP communications, Petitioner argues that Wookey discloses that client machine 10 has an “HTTP client agent,” such as a web browser, that can communicate using “any type of protocol.” Pet. 20 (quoting Ex. 1005 ¶ 159). For example, Wookey discloses that “client machine 10 includes a client agent which may be, for example, implemented as a software program,” such as “a Web Browser.” Ex. 1005 ¶ 159. Wookey further states that “[t]he client agent can use any type of protocol, such as a remote display protocol, and it can be, for example, an HTTP client agent.” Ex. 1005 ¶ 159. Petitioner cites Wookey’s disclosure that the connection between client machine 10 and remote machine 30’ can be over various protocols, including the Virtual Network Computing (VNC) protocol, which can run over various protocols including TCP/IP, IPX/SPX, and NetBEUI. Pet. 20 (citing Ex. 1005 ¶¶ 155, 216, 225). Petitioner argues that IPX/SPX and NetBEUI are non-TCP protocols. Pet. 20–21 (citing Ex. 1002 ¶¶ 127–128; Ex. 1013, 1:20–33). Petitioner also cites Wookey’s disclosure that connections between client machine 10 and remote machine 30 “can be

IPR2021-00629

Patent 10,075,564 B1

established using a variety of communication protocols,” such as “TCP/IP.”
 Pet. 53 (quoting Ex. 1005 ¶ 156).

(1) “Non-TCP”

With respect to the non-TCP communication, claim 24 recites,
 a network application that is configured to operate in accordance
 with a non-transmission control protocol (TCP) protocol that
 operates above an Internet Protocol (IP) layer and below a
 hypertext transfer protocol (HTTP) application layer, where the
 code, when used by the client node, results in the client node
 utilizing the network application to operate in accordance with
 the non-TCP protocol to
 perform three operations relating to establishing the non-TCP connection.
 Petitioner argues that, although Wookey discloses using non-TCP protocols,
 as discussed above, “*Wookey* does not expressly provide details regarding
 establishing and maintaining the second connection or the types of
 negotiable parameters used to establish the second connection.” Pet. 22
 (citing Ex. 1002 ¶ 133). Petitioner relies on Berg’s disclosures relating to
 RUDP in combination with Wookey to teach the non-TCP subject matter
 recited in claim 24. Pet. 22–50. Petitioner argues that the RUDP in the
 combination of Wookey and Berg would have operated between the IP and
 HTTP layers as taught by Berg’s Figure 2, in which RUDP layer 206 is
 above IP layer 202 and below Signaling Protocols Application(s) layer 208.
 Pet. 36–37 (citing Ex. 1007, 1:15–59, 5:27–55, 8:3–18, 8:24–31, Fig. 2;
 Ex. 1002 ¶ 161).

Petitioner provides various rationales to support the proposed
 combination of Wookey and Berg. Pet. 26–31. For example, Petitioner
 argues that a person of ordinary skill in the art would have “recognized the
 benefits of using the RUDP protocol described in Berg with Wookey’s
 system/method, such as providing client machine 10 with an improved

IPR2021-00629

Patent 10,075,564 B1

reliable connection to remote machine 30' with the versatility of negotiated timeout parameters consistent with Wookey's desired characteristics for the second connection." Pet. 28 (emphasis omitted). Petitioner also argues that the proposed "combination would have involved the use of known technologies (e.g., aspects of similar protocols) and design concepts and processes to obtain the foreseeable result of a reliable connection between Wookey's client machine 10 and remote machine 30'." Pet. 30 (citing Ex. 1002 ¶ 151).

The first operation recited for the non-TCP protocol in claim 24 is "receive, from a server node, a non-TCP packet during a setup of a non-TCP connection." Petitioner argues that Berg discloses that a client initiates a connection by sending a SYN segment to a server and that the server sends a SYN response. Pet. 38–39 (citing Ex. 1007, 17:42–46, 17:59–18:20, 18:56–62, 19:1–40, 24:40–47; Ex. 1002 ¶¶ 167–170). According to Petitioner, "the SYN response sent from remote machine 30' and received by client machine 10 in the Wookey-Berg combination is a non-TCP packet." Pet. 39 (citing Ex. 1002 ¶ 168).

The second operation recited for the non-TCP protocol in claim 24 is "identify metadata, that specifies a number of seconds or minutes, in an idle time period parameter field in the non-TCP packet, for an idle time period, where, as a result of a detection of the idle time period, the non-TCP connection is subject to deactivation." Petitioner argues that Berg discloses that the SYN segment contains negotiable parameters including a null segment timeout value and a transfer state timeout value, each of which teaches the idle time period parameter field, according to Petitioner.

Pet. 40–48 (citing Ex. 1007, 18:27–28, 18:58–60, 19:1–40, 20:36–47, 20:58–64, 21:15–23, 22:20–33, 23:45–24:4, 24:15–25, 24:44–48; Ex. 1002

IPR2021-00629

Patent 10,075,564 B1

¶¶ 173–194).

Petitioner argues that Berg’s “null segment timeout value indicates the amount of time to wait before sending a null segment when the connection has been idle (e.g., when a data segment has not been sent or received)” and, therefore, “represents an ‘idle time period,’ as claimed.” Pet. 43–44 (citing Ex. 1007, 20:36–41, 22:25–33; Ex. 1002 ¶ 180). Petitioner contends that, “if the null segment is not received or acknowledged, the connection is subject to being closed via an ‘auto reset.’” Pet. 45 (citing Ex. 1007, 23:45–49; Ex. 1002 ¶ 181). According to Petitioner, “[w]hen an auto-reset is necessary, either side of the connection can send a reset (RST) segment to close or reset a connection.” Pet. 45 (citing Ex. 1007, 18:27–28, 22:20–24, 23:60–24:4); *see* Ex. 1007, 22:20 (“The RST segment is used to close or reset a connection.”); *see also* Pet. 46–48 (discussing Berg’s transfer state timeout value).

The third operation recited for the non-TCP protocol in claim 24 is “determine, based on the metadata, a timeout attribute associated with the non-TCP connection.” Petitioner argues that Berg’s disclosure of setting the null segment timer based on the null segment timeout value teaches this subject matter. Pet. 49 (citing Ex. 1007, 20:36–38, 23:45–59; Ex. 1002 ¶ 197); *see also* Pet. 50 (setting forth contentions based on Berg’s “transfer state timeout value”).

(2) “TCP”

With respect to the TCP communication, claim 24 recites,

wherein the code, when used by the client node, results in the client node utilizing the network application to operate in accordance with the TCP protocol to perform a three-way TCP handshake with another server node for setting up a TCP

IPR2021-00629

Patent 10,075,564 B1

connection with the another server node that is different than the non-TCP connection.

Petitioner argues that “[a] three-way handshake method is necessarily performed to establish a TCP connection” because the TCP specification (RFC 793) requires a three-way handshake to establish a TCP connection. Pet. 54 (citing Ex. 1002 ¶¶ 207–208; Ex. 1008 (RFC 793), 31–32, 34–37); *see* Ex. 1008, 34 (“The ‘three-way handshake’ is the procedure used to establish a connection.”), 32 (“A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN’s.”).

(3) *Claims 25–27*

Claims 25, 26, and 27 recite additional subject matter regarding the TCP and non-TCP connections.

Claim 25 recites, “A system including the another server node which includes the non-transitory computer readable medium of claim 24, wherein the system is configured to communicate the code to the client node utilizing the TCP connection,⁷ for setting up the non-TCP connection with the server node.” Referring to its contentions for claim 24, Petitioner argues that Wookey discloses that remote machine 30 sends HTML page 288 having encoded URLs to client machine 10 over the TCP connection and that HTML page 288 is then used to establish the non-TCP connection with

⁷ Claim 25 appears to conflict with one aspect of claim 24. In particular, the “wherein” clause of claim 24, from which claim 25 depends, recites that “the code” is used by the client node “for setting up a TCP connection,” but claim 25 recites that “the TCP connection” is used “to communicate the code to the client node.” Thus, the code that is required to set up the TCP connection in the first place appears to be communicated to the client node only after the TCP connection is set up. The parties may wish to address this during the trial.

IPR2021-00629

Patent 10,075,564 B1

remote server 30'. Pet. 57–58 (citing Ex. 1005, Fig. 2C; Ex. 1002 ¶¶ 215–218).

Claim 26 recites, “The system of claim 25, wherein the system is configured to communicate the code to the client node via the TCP connection utilizing a hypertext transfer protocol (HTTP).” Petitioner argues that “remote machine 30 in the combined Wookey-Berg system is configured to communicate the code to client machine 10, via a HTTP-based web browser application 280, utilizing the TCP connection.” Pet. 59–60 (citing Ex. 1005 ¶¶ 155, 552; Ex. 1002 ¶¶ 219–224).

Claim 27 recites,

The system of claim 26, wherein the code, when used by the client node, results in the client node operating such that the non-TCP connection is setup between the client node and the server node instead of the TCP connection in order to permit communication, between the client node and the server node, of the timeout attribute, where the timeout attribute is not communicated when setting up the TCP connection between the system and the client node, but is communicated when establishing the non-TCP connection between the client and the server node.

Petitioner refers to its contentions for claim 24 regarding the non-TCP connection that is set up between client machine 10 and remote machine 30', which includes the timeout attribute (null segment timeout and transfer state timeout) exchange discussed with respect to Berg's RUDP, and Petitioner argues that the timeout attribute is not communicated during setup of the TCP connection. Pet. 61–63 (citing Ex. 1007, 18:56–21:35, 24:40–47; Ex. 1002 ¶¶ 226–232).

(4) Patent Owner's Arguments

Patent Owner disputes Petitioner's contentions for some of the “TCP” and “non-TCP” subject matter recited in claims 24–27. Prelim. Resp. 8–34.

IPR2021-00629

Patent 10,075,564 B1

As an initial matter, we note that Patent Owner refers to “Claim Requirement #1” and “Claim Requirement #2” but inconsistently identifies what these requirements are. For example, Patent Owner introduces the same language—“results in the client node ... setting up a TCP connection with the another server node that is different than the non-TCP connection”—as both Claim Requirement #1 and Claim Requirement #2. Prelim. Resp. 9–10, 12–13. Later, Patent Owner argues that “claim 27 requires that ‘the code, when used by the client node, results in the client node operating such that the non-TCP connection is setup between the client node,’ *i.e.*, Claim Requirement #2.” Prelim. Resp. 15. Based on our review of Patent Owner’s arguments, we understand Patent Owner to be arguing that Petitioner has not shown that Wookey teaches code that results in the client node establishing a non-TCP connection with the server node and a TCP connection with the another server node. Prelim. Resp. 8–34.

One of Patent Owner’s arguments is that Petitioner relies on different embodiments—Figs. 3D and 25—of Wookey to teach the TCP and non-TCP communications. Prelim. Resp. 10–16. According to Patent Owner, “*Wookey*’s URL/HTML do **not** provide the claimed nexus between the Claim Requirement #1 and #2,” and “a particular ‘address’ in *Wookey*’s Second Embodiment (Fig. 25) is disclosed to be separate from (and has no nexus to) the URL in *Wookey*’s First Embodiment (Fig. 3D).” Prelim. Resp. 14. Patent Owner argues, therefore, that “in the Second Embodiment *Wookey* discloses, that connection establishment via a different protocol results when there is use of the ‘address’ retrieved by the client machine from the remote machine 30, and **not** *Wookey*’s ‘fetched’ URL/HTML.” Prelim. Resp. 14.

On this record, we disagree with Patent Owner’s arguments. Petitioner cites paragraph 732 of Wookey to show that client machine 10 can communicate with remote machine 30 using a protocol that is different from the one used for communications with remote machine 30’. *See* Pet. 21–22 (citing Ex. 1005 ¶ 732). As an example of this, Wookey explains that “client machine 10 may make a request to the remote machine 30 using the IPX protocol and request the address of the remote machine 30’ as a TCP/IP protocol address.” Ex. 1005 ¶ 732. But this is not the only disclosure in Wookey on which Petitioner relies. Indeed, in addressing the “wherein” clause of claim 24, which recites “a TCP connection with the another server node that is different than the non-TCP connection,” Petitioner relies on Wookey’s disclosure in paragraph 156 that the link between client machine 10 and remote machine 30 can be “TCP/IP.” Pet. 53 (citing Ex. 1005 ¶ 156); *see also* Pet. 53 n.21 (citing Ex. 1005 ¶¶ 155, 159, 216, 225). Wookey explains that “communications link 150 may provide communications functionality through a variety of connections,” which “can be established using a variety of communication protocols (e.g., TCP/IP[]).” Ex. 1005 ¶ 156. Communications link 150 is between client machine 10 and remote machine 30. Ex. 1005 ¶ 154 (“[T]he client machine 10 communicates with the remote machine 30 in the machine farm 38 over a communications link 150.”). On this record, we are sufficiently persuaded that Wookey’s express disclosure of TCP/IP communications between client machine 10 and remote machine 30 teaches a client node “setting up a TCP connection with the another server node that is different than the non-TCP connection,” as recited in the wherein clause of claim 24.

On this record, we also disagree with Patent Owner’s arguments that the “code” identified by Petitioner does not “result[] in” the establishment of

IPR2021-00629

Patent 10,075,564 B1

the TCP and non-TCP connections. *See* Prelim. Resp. 13–16. Rather, on this record, we are sufficiently persuaded by Petitioner’s contentions, summarized above, that HTML page 288, when used by client machine 10, results in connections being made to remote machines 30 and 30’ and that these connections can be TCP and non-TCP connections. *See* Pet. 19–20 (discussing HTML page 288 as “code”), 20–22 (discussing client machine 10’s use of HTML page 288 to set up a non-TCP connection between client machine 10 and remote machine 30’), 53 (discussing client machine 10’s use of HTML page 288 to set up a TCP connection between client machine 10 and remote machine 30). For example, Wookey explains that “HTML page 288 . . . includes a Resource Neighborhood window 258 in which appears graphical icons 257, 257’ representing resources to which the client machine 10 has access. A user of client machine 10 requests access to a resource represented by icon 257 by clicking that icon 257.” Ex. 1005 ¶ 196. Thus, on this record, we are persuaded that HTML page 288 is code that, when executed by a client node, results in the client node making connections to remote resources. Furthermore, Wookey discloses that the connection between client machine 10 and remote machine 30 can be “TCP/IP” and that the connection between client machine 10 and remote machine 30’ can be IPX/SPX and NetBEUI, i.e., “non-TCP” protocols. Ex. 1005 ¶¶ 156, 225; Ex. 1002 ¶ 127 (Dr. Lin testifying that “[a] person of ordinary skill in the art would have understood that the IPX/SPX and NetBEUI protocols, for example, are non-TCP-based protocols, as claimed.”).

c) Determination for Claims 25–27

On this record, we are sufficiently persuaded by Petitioner’s contentions and evidence, summarized above, and we disagree with Patent

IPR2021-00629

Patent 10,075,564 B1

Owner's arguments. In particular, we are persuaded that Petitioner has explained sufficiently how the proposed combination of Wookey and Berg teaches the subject matter recited in claims 25–27 and that a person of ordinary skill in the art would have had reason to combine the teachings of Wookey and Berg in the manner asserted. Having considered the parties' arguments, we determine that Petitioner has established a reasonable likelihood of prevailing in challenging claims 25–27 as obvious over the combined teachings of Wookey and Berg.

III. CONCLUSION

For the foregoing reasons, we determine that the information presented in the Petition establishes that there is a reasonable likelihood that Petitioner would prevail in challenging at least one claim of the '564 patent, and we institute *inter partes* review of all remaining challenged claims (claims 25–27) on the sole ground raised in the Petition. *See* PTAB Rules of Practice for Instituting on All Challenged Patent Claims and All Grounds and Eliminating the Presumption at Institution Favoring Petitioner as to Testimonial Evidence, 85 Fed. Reg. 79,120, 79,129 (Dec. 9, 2020) (codified at 37 C.F.R. 42.108(a)). We do not address claims 24, 28, and 29 because Patent Owner disclaimed them. *See* Prelim. Resp. 1 (“Jenam statutorily disclaimed claims 24, 28, and 29. Only claims 25-27 remain.”). At this stage of the proceeding, we have not made a final determination with respect to the patentability of any of the challenged claims or the construction of any claim term.

IPR2021-00629

Patent 10,075,564 B1

IV. ORDER

Accordingly, it is

ORDERED that, pursuant to 35 U.S.C. § 314(a) and 37 C.F.R. § 42.4, an *inter partes* review is hereby instituted as to claims 25–27 of the ’564 patent and on all challenges raised in the Petition; and

FURTHER ORDERED that, pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial, which will commence on the entry date of this decision.

IPR2021-00629

Patent 10,075,564 B1

FOR PETITIONER:

Naveen Modi

Joseph Palys

Quadeer Ahmed

Jason Heidemann

PAUL HASTINGS LLP

naveenmodi@paulhastings.com

josephpalys@paulhastings.com

quadeerahmed@paulhastings.com

jasonheidemann@paulhastings.com

FOR PATENT OWNER:

Timothy Devlin

DEVLIN LAW FIRM LLC

td-ptab@devlinlawfirm.com

EXHIBIT 16

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

JENAM TECH, LLC,
Patent Owner.

IPR2021-00630
Patent 10,075,565 B1

Before DANIEL J. GALLIGAN, SCOTT B. HOWARD, and
JASON M. REPKO, *Administrative Patent Judges*.

REPKO, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

IPR2021-00630

Patent 10,075,565 B1

I. INTRODUCTION

A. Background

Google LLC (“Petitioner”) filed a Petition requesting *inter partes* review of claims 25 and 28 of U.S. Patent No. 10,075,565 B1 (“the ’565 patent,” Ex. 1001). Paper 1 (“Pet.”). Jenam Tech, LLC (“Patent Owner”) filed a Preliminary Response. Paper 6 (“Prelim. Resp.”). According to Patent Owner, claim 25 has been disclaimed, and only claim 28 remains in this proceeding. *Id.* at 1.

With our authorization, Petitioner filed a Reply to address discretionary-denial issues (Paper 8), and Patent Owner filed a Sur-reply (Paper 9).

Under 37 C.F.R. § 42.4(a), we have authority to determine whether to institute review. An *inter partes* review may not be instituted unless the information presented in the Petition and the Preliminary Response shows “there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a).

For the reasons explained below, we institute an *inter partes* review.

B. Related Matters

As required by 37 C.F.R. § 42.8(b)(2), the parties identify various related matters. Pet. 1–2; Paper 4 (Patent Owner’s Mandatory Notice).

C. The ’565 Patent

The *Background* section of the ’565 patent describes the problem that is purportedly addressed by the disclosed system as follows. *See* Ex. 1001, 1:44–2:29.

IPR2021-00630

Patent 10,075,565 B1

Some TCP implementations support a keep-alive option, which is not part of the TCP specification described in RFC¹ 793. *Id.* at 1:48–56. Critics of the keep-alive option argue that it wastes network bandwidth. *Id.* at 1:62–65. Critics also argue that, because the TCP-connection endpoints do not cooperate in supporting the keep-alive option, the nodes oppose one another or waste resources by duplicating function. *Id.* at 1:65–2:3.

The option’s proponents, though, argue that there are benefits to keeping an inactive TCP connection open. *Id.* at 2:9–11. For example, the keep-alive option prevents firewalls from closing idle or inactive TCP connections to recover resources. *Id.* at 2:11–13. Keep-alive packets might cause the firewall to waste resources and possibly block or terminate TCP connection with other nodes. *Id.* at 2:14–18.

According to the ’565 patent, there is no mechanism that allows two TCP-connection endpoints to cooperate in supporting the keep-alive option. *Id.* at 2:20–22. The ’565 patent purportedly addresses the need to share information for detecting an idle TCP connection. *Id.* at 2:26–29.

Claims 25 and 28 are reproduced below.

25. An apparatus comprising:
a server computer including:
a non-transitory memory storing instructions; and
one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions such that a network application operates in accordance with a first protocol including a transmission control protocol (TCP), the server computer, when operating in accordance with the first protocol to set up a TCP connection, configured to:
communicate a segment including at least one first synchronize bit;

¹ RFC stands for “Request for Comments.”

IPR2021-00630

Patent 10,075,565 B1

communicate a first acknowledgement of the segment, and at least one second synchronize bit; and

communicate a second acknowledgement;

said server computer further configured to communicate, to a client computer, code that, when used by the client computer, causes the client computer to operate in accordance with a second protocol that is different from the TCP and that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, in order to setup a second protocol connection with another server computer, by: receiving, by the client computer from the another server computer, a packet;

identifying metadata, that specifies a number of seconds or minutes, in an idle time period parameter field in the packet for an idle time period, during which, no packet is communicated that meets each of the following criteria:

a) communicated via the second protocol connection, and

b) causes the second protocol connection to be kept at least partially alive; and

determining, by the client computer and based on the metadata, a timeout attribute associated with the second protocol connection.

Ex. 1001, 26:37–27:4.

28. The apparatus of claim 25 wherein the code is communicated by permitting the client computer to fetch the code from the server computer utilizing the TCP connection and a hypertext transfer protocol (HTTP), and further wherein the code, when used by the client computer, causes the client computer to operate in accordance with the second protocol that is different from the TCP, by:

detecting the idle time period based on the timeout attribute; and

deactivating the second protocol connection by communicating a particular packet between the client computer and the another server computer, in response to detecting the idle

IPR2021-00630

Patent 10,075,565 B1

time period.

Id. at 27:63–28:8.*D. Asserted Ground of Unpatentability*

Petitioner challenges the patentability of claim 28 of the '565 patent on the following ground:

Claim(s) Challenged	35 U.S.C. § ²	Reference(s)/Basis
28	103(a)	Wookey, ³ Berg ⁴

Petitioner also relies on the Declaration of Dr. Bill Lin. Ex. 1002 (“Lin Decl.”).

II. ANALYSIS

*A. Discretionary Denial**1. 35 U.S.C. § 314(a)*

Patent Owner argues that we should exercise discretion to deny the Petition under 35 U.S.C. § 314(a) based on the factors set forth in *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 (PTAB Sept. 6, 2017) (§ II.B.4.i designated as precedential on Oct. 17, 2017). Prelim. Resp. 54–68.

According to Patent Owner, “[t]his Petition is *one of two* that Google filed against [the] '565 patent.” *Id.* at 55. Patent Owner provides a table that purportedly “summarizes the various petitions, challenged patents and claims, and asserted references, and illustrates the *substantial overlap* that

² The Leahy-Smith America Invents Act (“AIA”) included revisions to 35 U.S.C. § 103 that became effective March 16, 2013. The application for the '565 patent was filed on March 7, 2018. Petitioner asserts that the claims are unpatentable under pre-AIA § 103(a). Pet. 3–4.

³ Wookey, US 2007/0171921 A1, published July 26, 2007 (Ex. 1005).

⁴ Berg, US 6,674,713 B1, published Jan. 6, 2004 (Ex. 1007).

IPR2021-00630

Patent 10,075,565 B1

exists across the various proceedings.” *Id.* at 55–56. The table lists *three* IPRs allegedly involving the ’565 patent: IPR2021-00628, IPR2021-00629, and IPR2021-00630. *Id.* But the petitions filed in IPR2021-00628 and IPR2021-00629 challenge Patent 10,075,564, not the ’565 patent. Patent Owner does not identify any other petition challenging the claims of the ’565 patent. *See id.*

General Plastic, on the other hand, sets forth a series of factors to be considered by the Board in evaluating whether to exercise discretion under § 314(a) to deny a petition that challenges a patent that was previously challenged before the Board. *General Plastic*, Paper 19 at 15–19. As *General Plastic* states, “[m]ultiple, staggered petitions *challenging the same patent and same claims* raise the potential for abuse.” *Id.* at 17 (emphasis added). But that is not the case here because the ’565 patent has only been challenged once. Patent Owner’s assertions about multiple petitions challenging the ’565 patent are unfounded. Prelim Resp. 55. Thus, Patent Owner has not shown that the “potential for abuse” from multiple petitions identified in *General Plastic* is present here.

Patent Owner also argues that “the Board has previously denied institution when related, follow-on petitions challenged claims similar in scope and content, even when the claims were in different patents.” *Id.* at 60 (citing *Tietex Int’l, LTD., v. Precision Fabrics Grp., Inc.*, IPR2015-01671, Paper 7, at 8 (PTAB Feb. 11, 2016)). According to Patent Owner, “[t]he Board should similarly exercise its discretion to deny this Petition. Otherwise [Patent Owner] would be forced to expend resources defending claims that are similar in scope and content to those of [U.S. Patent 9,923,995 B1 (“the ’995 patent”)] it is already defending.” *Id.* at 60–61.

IPR2021-00630

Patent 10,075,565 B1

The decision denying institution in *Tietex*, which is not precedential,⁵ is readily distinguishable from the Petition here. In *Tietex*, the panel denied institution on one ground⁶ in which the petition presented “the same arguments that we[re] found to be unpersuasive in” a previous final written decision addressing claims of similar scope in a related patent. *Tietex*, Paper 7 at 8. By contrast, we have issued a final written decision in IPR2020-00845 holding all challenged claims of the ’995 patent unpatentable. *See Google LLC v. Jenam Tech., LLC*, IPR2020-00845, Paper 33 (PTAB Aug. 27, 2021). Thus, even assuming that Patent Owner is correct that claim 28 of the ’565 patent is similar in scope to any of the claims of the ’995 patent, the reasoning in *Tietex* is inapplicable to the facts here.

For these reasons, we do not deny the Petition based on *General Plastic* under § 314(a).

2. 35 U.S.C. § 325(d)

Patent Owner argues that the Petition should be denied under 35 U.S.C. § 325(d) because Berg was previously presented to the Office in IPR2020-00742 (“the ’742 IPR”) and the Petition’s arguments about Berg are “substantially similar” to those in the ’742 IPR. Prelim. Resp. 68 (citing *Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH*,

⁵ “A precedential decision is binding Board authority in subsequent matters involving similar facts or issues.” Patent Trial and Appeal Board, Standard Operating Procedure 2, Rev. 10 (Sept. 20, 2018), <https://www.uspto.gov/sites/default/files/documents/SOP2%20R10%20FINAL.pdf>.

⁶ The *Tietex* decision issued before the Supreme Court’s holding that a decision to institute under 35 U.S.C. § 314 may not institute on fewer than all claims challenged in the petition. *See SAS Inst., Inc. v. Iancu*, 138 S. Ct. 1348, 1359–60 (2018).

IPR2021-00630

Patent 10,075,565 B1

IPR2019-01469, Paper 6, 7–8 (PTAB Feb. 13, 2020) (precedential)). We disagree with Patent Owner’s argument because the ’742 IPR involved a different patent (the ’995 patent), and our precedent states that, “[u]nder § 325(d), the art and arguments must have been previously presented to the Office during proceedings pertaining to *the challenged patent*.” *Advanced Bionics*, Paper 6 at 7 (emphasis added). Thus, we do not deny the Petition under § 325(d) based on the use of Berg in the ’742 IPR.⁷

B. Patent Owner’s Arthrex Argument

Patent Owner argues that, based on the Supreme Court’s decision in *United States v. Arthrex, Inc.*, 141 S.Ct. 1970 (2021), “the Board does not have authority to decide issues of patentability presented in a petition for inter partes review” until procedures are put in place by which the Director can review decisions by Administrative Patent Judges. Prelim. Resp. 68–69.

The Court in *Arthrex* held that “Congress has vested the Director with the ‘power and duties’ of the PTO, § 3(a)(1),” and that “[t]he Director accordingly may review final PTAB decisions and, upon review, may issue decisions himself on behalf of the Board.” *Arthrex*, 141 S. Ct. at 1987. Because the Director has the authority to review final decisions of the Board, we disagree with Patent Owner’s suggestion that the absence of specific review procedures results in a Constitutional infirmity.

Also, Patent Owner’s argument is moot because the Office has implemented an interim Director-review process following *Arthrex*:

Parties may request Director review of a final written decision in an inter parties review or a post-grant review by concurrently (1) entering a Request for Rehearing by the Director into PTAB

⁷ The ’742 IPR was terminated based on a settlement between the parties to that proceeding before a final written decision issued. IPR2020-00742, Paper 31.

IPR2021-00630

Patent 10,075,565 B1

E2E and (2) submitting a notification of the Request for Rehearing by the Director to the Office by email to Director_PTABDecision_Review@uspto.gov, copying counsel for all parties by email.

See USPTO implementation of an interim Director review process following *Arthrex* (Jun. 29, 2021), <https://www.uspto.gov/patents/patent-trial-and-appeal-board/procedures/uspto-implementation-interim-director-review>.

Thus, we decline to deny institution based on the Supreme Court’s decision in *Arthrex*.

C. Claim Construction

We interpret claim terms using “the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. § 42.100(b) (2019).

1. Idle Time Period (“ITP”)

Claim 28, through its incorporation of the limitations of claim 25 from which it depends, recites “an idle time period, during which, no packet is communicated that meets each of the following criteria: a) communicated via the second protocol connection, and b) causes the second protocol connection to be kept at least partially alive.” Ex. 1001, 26:64–27:2. We refer to this as the “ITP limitation.”

Patent Owner’s proposed construction for the ITP limitation is summarized below with Patent Owner’s formatting.

Claim Language	Patent Owner’s Proposed Construction
“ <i>idle time period</i> , during which, <i>no packet</i> is <i>communicated</i> that meets	“ <i>idle time period</i> , during which, no packet <i>of any kind</i> is <i>sent or</i>

IPR2021-00630

Patent 10,075,565 B1

each of the following criteria: a) communicated via the second protocol connection, and b) <i>causes the second protocol connection to be kept at least partially alive</i>	<i>received</i> that meets each of the following criteria: a) communicated via the second protocol connection, and b) causes the second protocol connection to be kept at least partially alive <i>with or without a condition and via any mechanism;</i>
--	---

Prelim. Resp. 8–9; *see also id.* at 8–24 (presenting arguments about these constructions). As in the table above, Patent Owner’s proposed construction is based on the following three terms or phrases in the recited subject matter: “no packet,” “communicated,” and “causes the second protocol connection to be kept at least partially alive.” *Id.* at 8–24.

As to the language “no packet,” Patent Owner argues that there is no “modifier” for “no packet” and, thus, that “the ‘no packet’ limitation is ***unqualified***, prohibiting ***all*** types of packets from being communicated to keep the TCP-variant connection active.” *Id.* at 10.⁸ Based on the current record, we disagree. The claim recites two criteria for the communicated packet: (1) the packet must be “communicated via the second protocol connection,” and (2) must cause “the second protocol connection to be kept at least partially alive.” Ex. 1001, 26:64–27:2. Thus, the phrase “no packet of any kind” is qualified by the recited criteria.

Patent Owner argues that the term “communicated” means “sent or received by the system.” Prelim. Resp. 13–24. According to Patent Owner, “since the claims refer to the connection in which the system is a part, ‘communicated’ refers to packets being sent or received by that system.”

⁸ Patent Owner refers to the “TCP-variant connection.” *See, e.g.*, Prelim. Resp. 10. But the claim does not recite such a connection. Rather, claim 28 recites a “second protocol connection.” Ex. 1001, 26:59–60.

IPR2021-00630

Patent 10,075,565 B1

Id. at 13. Stated differently, Patent Owner argues that “‘communicated’ should not be limited to **only** [‘]receiving,’ ‘sending,’ or ‘receiving **and** sending.’ It is broader. Encompassing sending **or** receiving is consistent with the claim’s scope and claim differentiation.” *Id.* at 16.

As explained below, we determine on this record that Petitioner has shown that the asserted prior art teaches “communicated” under Patent Owner’s proposed construction. So, for purposes of this Decision, we apply Patent Owner’s construction of “communicated,” and we need not resolve any disputes between the parties over the scope of this term. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (“[W]e need only construe terms ‘that are in controversy, and only to the extent necessary to resolve the controversy’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

Patent Owner argues that the phrase “causes the second protocol connection to be kept at least partially alive” means “causes the second protocol connection to be kept at least partially alive with or without a condition and via any mechanism.” Prelim. Resp. 9, 22–24. As explained below, we find that the asserted prior art teaches, in at least one scenario, an absence of packets that are communicated to keep the connection active. So we apply Patent Owner’s construction of the phrase “causes the second protocol connection to be kept at least partially alive” and need not resolve any disputes regarding the scope of this phrase.

For purposes of this Decision, we need not construe expressly any other claim terms. *See Nidec*, 868 F.3d at 1017.

IPR2021-00630

Patent 10,075,565 B1

D. Level of Ordinary Skill in the Art

Petitioner argues that a person having ordinary skill in the art “would have had an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking.” Pet. 4 (citing Ex. 1002 ¶¶ 16–18). Petitioner further argues that “[m]ore education can supplement practical experience and vice versa.” *Id.*

Patent Owner substantially agrees with Petitioner’s proposal. *See* Prelim. Resp. 5.

So, for this Decision, we adopt Petitioner’s proposed formulation of the level of ordinary skill in the art, except that we delete the qualifier “at least” to eliminate vagueness as to the amount of practical experience. The qualifier expands the range indefinitely without an upper bound, and thus precludes a meaningful indication of the level of ordinary skill in the art.

E. Obviousness

1. Overview of the Challenge to Claim 28

Claim 28 depends from and incorporates all the subject matter recited in disclaimed independent claim 25. In its Preliminary Response, Patent Owner contests limitations directed to the code (“the code limitation”) and the idle time period (“the ITP limitation”) found in claim 25. Prelim. Resp. 24–54. We discuss Patent Owner’s arguments and those limitations in the sections that follow. *Infra* §§ II.E.2, 3.

As for the remaining limitations, we have reviewed Petitioner’s assertions and the cited evidence and preliminarily determine that Petitioner has sufficiently shown—at this stage and on this record—that the subject matter recited would have been obvious. *See* Pet. 10–70. In the rest of this section, we provide an overview of the Petition’s analysis of Wookey, Berg,

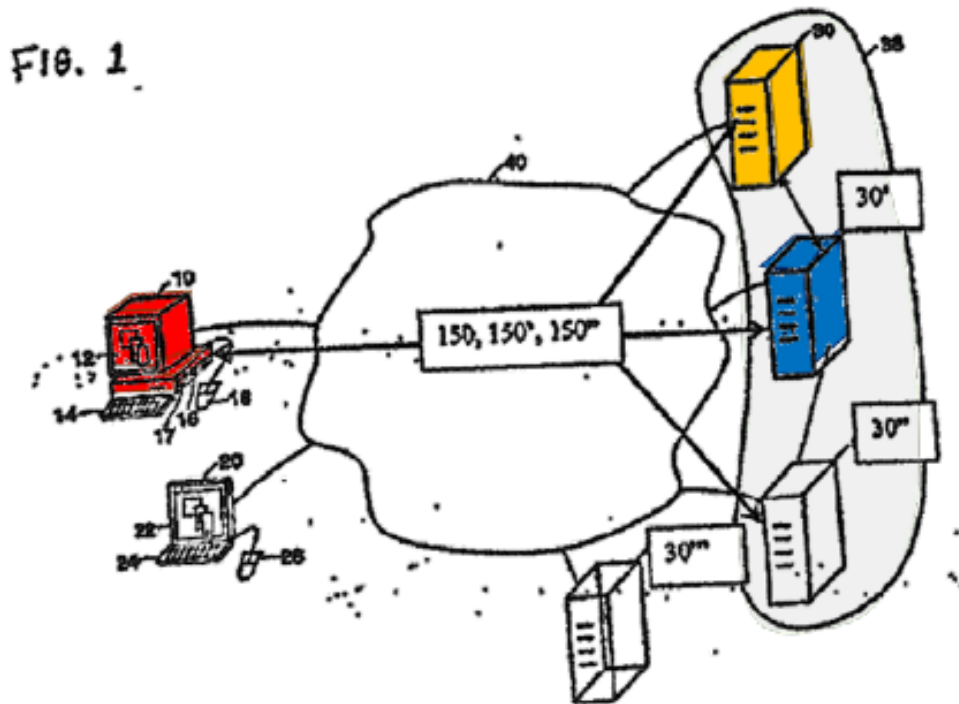
IPR2021-00630

Patent 10,075,565 B1

and claim 28.

Petitioner asserts that Wookey teaches the subject matter recited in claim 28, except that “*Wookey* does not expressly provide details regarding establishing and maintaining the second connection or the types of negotiable parameters used with those protocols for the second connection.” Pet. 27. Petitioner relies on Berg in combination with Wookey for the subject matter pertaining to the second protocol. *See id.* at 28–70.

Wookey generally relates to providing access to computing environments. Ex. 1005 ¶ 2. Wookey discloses an environment in which a client machine can access computing resources provided by remote machines. *Id.* ¶ 135. Petitioner provides the following annotated version of Wookey’s Figure 1 to show various components recited in claim 28:



Pet. 11. Wookey’s Figure 1, reproduced above with color added by Petitioner, “is a block diagram of one embodiment of an environment in which a client machine accesses a computing resource provided by a remote

IPR2021-00630

Patent 10,075,565 B1

machine.” Ex. 1005 ¶ 20. In the annotated version of Wookey’s Figure 1, above, client machine 10 is red, remote machine 30 is orange, and remote machine 30’ is blue. Pet. 11. Petitioner maps 10, 30, and 30’ in Figure 1 to “a client node,” “another server node,” and “a server node” in claim 28. *Id.* at 20–22.

Wookey teaches an embodiment with an HTML page (288) that displays images of computing environments that are available to a client machine. Ex. 1005 ¶ 213. In this embodiment, a user of client machine 10 clicks on icons corresponding to the resources displayed on HTML page 288 to establish a connection to the remote machines shown in Figure 1, above. *Id.* ¶¶ 26, 213–216, *cited in* Pet. 21–22.

To address the details of the recited “second protocol,” Petitioner cites Berg’s teachings about Reliable User Datagram Protocol (RUDP). Pet. 38–39. When using RUDP, hosts negotiate various parameters. Ex. 1007, 17:9–17, 18:56–19:40. In particular, Berg uses a synchronization (SYN) segment to establish a connection and synchronize sequence numbers between two hosts. *Id.* at 18:57–58. The SYN segment contains various negotiable parameters. *Id.* at 18:58–60, 20:37–38, 20:43–45.

Of those parameters, Petitioner’s obviousness rationale relies upon Berg’s “Null Segment Timeout Value,” which specifies “[t]he timeout value for sending a null segment if a data segment has not been sent” (*id.* at 20:36–37), and a “Transfer State Timeout Value,” which specifies “the amount of time the state information will be saved for a connection after an auto reset occurs” (*id.* at 20:43–45). Pet. 47–49. Berg’s client sends a null segment to the server if the null-segment timeout period elapses without a data segment having been sent. Ex. 1007, 23:48–49.

Under Petitioner’s proposed combination, “the code in the Wookey-

Berg combination causes client machine 10 (e.g., by way of its web browser application 280) to operate in accordance with the RUDP when establishing a connection between client machine 10 and remote machine 30’.” Pet. 38–39. We next turn to the issues raised in Patent Owner’s Preliminary Response regarding that code.

2. *The Code Limitation*

Claim 28 requires, through its dependency on claim 25, a system including a “server computer further configured to communicate, to a client computer, code that, when used by the client computer, causes the client computer to . . . to setup a second protocol connection with another server computer.” We refer to this as the “code limitation.” See Prelim. Resp. 24.

Petitioner argues that Wookey teaches the code limitation in two ways. See Pet. 21–23, 24 n.12. First, Petitioner asserts that Wookey’s “encoded URLs are the claimed ‘code’ because such encoded URLs are instructions that, when used by client machine 10, command client machine 10 to access resources from one or more locations (e.g., one or more remote machines).” *Id.* at 22 (citing Ex 1002 ¶ 130) (emphasis omitted). Second, Petitioner asserts that Wookey’s “entire HTML page 288 (including the encoded URLs and icons) also discloses the claimed ‘code . . . used by the client computer,’ because an HTML page was known to comprise HTML code used to define, e.g., what is displayed on the screen of the client node.” *Id.* at 23 (citing Lin Decl. ¶¶ 61–62, 132; Ex. 1012 ¶¶ 22, 24, 26, 79).

Patent Owner’s arguments refer to “URL/HTML” in describing Petitioner’s challenge. See, e.g., Prelim. Resp. 30. Apart from these references to “UML/HTML,” Patent Owner does not substantively address why Wookey’s HTML page does not contain the recited code. See *id.* Rather, Patent Owner argues that Wookey’s URL is an address, not code.

IPR2021-00630

Patent 10,075,565 B1

Id. at 28. For example, Patent Owner argues that “‘code’ cannot read on a URL address *by itself*.” *Id.* at 29. Wookey’s encoded URL is also the basis for Patent Owner’s other arguments.⁹ *See id.* at 28–37.

On this record and at this preliminary stage, we are sufficiently persuaded by Petitioner’s explanation that Wookey’s “entire HTML page 288 (including the encoded URLs and icons)” teaches the recited code because the page has code used to define what is displayed on the client’s screen. Pet. 23 (citing Lin Decl. ¶ 132). Also, Dr. Lin’s testimony provides the basis for his opinion that an HTML page is code. *See* Lin Decl. ¶ 132 (citing Ex. 1012 ¶¶ 22, 24, 26, 79). Thus, to institute here, we need not rely on Petitioner’s alternative rationale that Wookey’s URL is “code.” *See* Pet. 21–23.

Patent Owner further argues that Petitioner improperly combined distinct embodiments from Figures 3 and 25 of Wookey. *See* Prelim. Resp. 25–28, 38–39. In Patent Owner’s view, Figure 25 uses a different connection protocol than Figure 3. *Id.*

Based on the current record, we disagree that the Petition is flawed in this way. Rather, we preliminarily determine that Wookey’s Figures 3D and 25 show different aspects of the same environment: Both figures show how client machine 10 interacts with remote machine 30 to access a resource. *See* Ex. 1005, Figs. 3D, 25. Figure 3D focuses on how the “client machine can access a resource from a resource neighborhood web page displayed at that client machine,” whereas Figure 25 depicts “a remote machine and

⁹ Patent Owner also argues that Wookey’s address is not code. Prelim. Resp. 41. Yet Patent Owner acknowledges that Petitioner has not made this argument. *Id.* at 37. Thus, we do not discuss Patent Owner’s reasons why arguments based on Wookey’s address would fail. *See id.* at 37–44.

IPR2021-00630

Patent 10,075,565 B1

client machine establishing a protocol stack for communication.” *Id.* ¶¶ 26, 57. Thus, in these figures, Wookey teaches different aspects of communication within the environment shown in Figure 1. *See id.* ¶ 135 (discussing environment containing client machine 10 and remote machine 30).

Even if we assume that Wookey’s figures show different embodiments, we would still disagree with Patent Owner’s argument because it mischaracterizes Petitioner’s reliance on Figure 25. *See Prelim. Resp.* 25–28. Petitioner is not relying on a specific communication feature from Figure 25. Rather, Petitioner is relying on Wookey’s description of the figure to support the assertion that different communication protocols can be used. *See, e.g.,* Pet. 25, 27 (citing Ex. 1005 ¶ 732).

As for that assertion, Petitioner has sufficiently shown that Wookey teaches using a different connection protocol between client machine 10 and remote machine 30’ than was used initially between client machine 10 and server node/remote machine 30. *See* Ex. 1005 ¶¶ 731–732 (describing Figure 25). Wookey teaches that, although client machine 10 and remote machine 30 may use a TCP/IP protocol, it may use the IPX protocol to connect with remote machine 30’. *Id.* ¶ 732. At this stage, Petitioner’s reliance on these teachings is sufficient to support the obviousness rationale based on Wookey and Berg. *See* Pet. 27. We analyze Berg’s specific communication features in the next section.

3. *The ITP Limitation*

Claim 28, through its dependency on claim 25, recites an “idle time period, during which, no packet is communicated that meets each of the following criteria: a) communicated via the second protocol connection, and b) causes the second protocol connection to be kept at least partially alive.”

IPR2021-00630

Patent 10,075,565 B1

We refer to this limitation as the “ITP limitation,” as Patent Owner does.

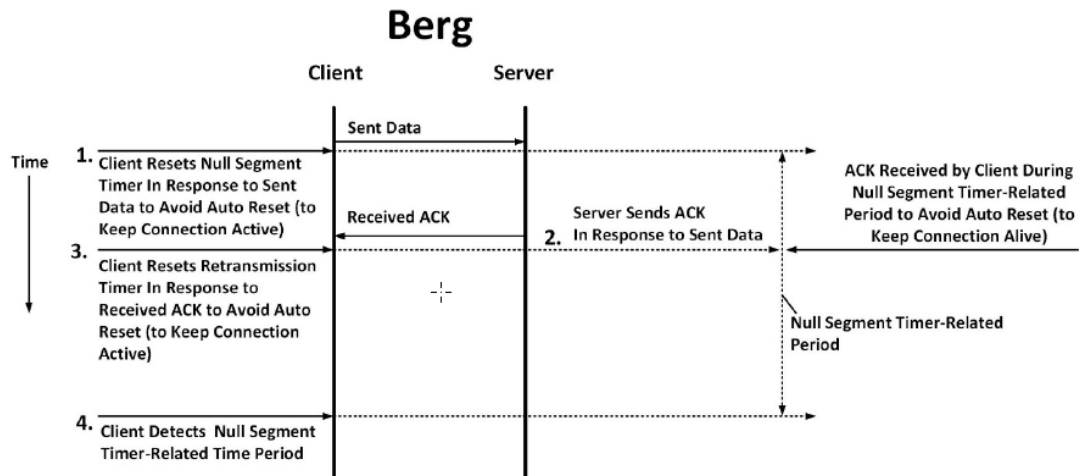
See Prelim. Resp. 45.

Petitioner provides three alternative rationales to address the ITP limitation: First, Petitioner asserts that Berg’s “period of waiting by the client (when the connection is idle) before the client sends a null segment to the server to verify whether the connection is active represents an ‘idle time period,’ as claimed.” Pet. 47. Second, Petitioner asserts Berg’s “transfer state timeout value (‘metadata’), which is specified in a number of seconds, is identified from a transfer state timeout value parameter field (‘idle time period parameter field’) in an RUDP SYN segment.” *Id.* at 48–49. Third, Petitioner asserts “idle time periods discussed above (based on the null segment timeout value and the transfer state timeout value) satisfy [the ITP limitation] in an additional way by the Wookey-Berg combination where a disconnection occurs between the client machine 10 and remote machine 30’.” *Id.* at 56.

Although Petitioner presents three alternative rationales, Patent Owner’s arguments largely focus on Petitioner’s first rationale based on the null segments. Prelim. Resp. 45–54. In particular, Patent Owner argues that, during the null segment timer-related period, Berg sends ACKs, which are intended to keep the connection alive. *Id.* at 50. Patent Owner illustrates this argument with a diagram, reproduced below. *Id.*

IPR2021-00630

Patent 10,075,565 B1



Patent Owner’s timing diagram, above, shows a client sending data to a server, and a server responding with an ACK. *Id.*

This argument, however, does not address the situation in which a disconnection occurs between the client machine 10 and remote machine 30’, as in Petitioner’s disconnection hypothetical. Pet. 56. Rather, on this record and at this stage, Petitioner has sufficiently shown that “when an interruption in communication occurs, such as a disconnection in the network service (as contemplated by *Wookey* and *Berg*) or in a physical network medium (as described in the ’565 patent) (EX1001, 2:5–8, 2:23–26, 12:37–43), packets from one node do not reach the other node in either direction.” *Id.*

Contrary to Patent Owner’s conclusory argument (Prelim. Resp. 46), Petitioner’s assertions are adequately supported for the purpose of institution: Petitioner asserts that both *Wookey* and *Berg* “aim to maintain connection between nodes through unintentional disconnection periods.” Pet. 56. This is discussed at least in Section IX.A.1.e.1.b. of the Petition. *See id.* In that section, Petitioner supports the obviousness rationale by citing the Lin Declaration, as well as *Wookey*’s and *Berg*’s discussion of handling disruptions and maintaining connections. *Id.* at 33 (citing

IPR2021-00630

Patent 10,075,565 B1

Lin Decl. ¶¶ 20–21, 155; Ex. 1005 ¶¶ 581, 1135–1136, 1153; Ex. 1007, 1:16–32, 1:54–2:11, 2:43–47).

Through citation and discussion of Wookey and Berg, Dr. Lin’s testimony provides a sufficient basis for his opinions about establishing a reliable connection between two nodes. Lin Decl. ¶¶ 20–21, 155. In particular, the cited passage from Berg states that “[t]o provide reliable backhaul, it is important to be able to maintain multiple IP connections or sessions between the signaling terminal device and the call processing device, so that message[s] can be transmitted even when the inherently unreliable IP network fails,” and that “[t]he system must be configured so that communications are not interrupted upon network failure, and so that communications can resume when the network comes back up.” Ex. 1007, 2:4–11, *quoted in* Lin Decl. ¶ 155. Likewise, the cited passage from Wookey states that “the user and/or the client machine 10 may traverse network segments or network access points that cause changes in the network address or host name, e.g., internet protocol (IP) address, of the client machine 10 or causes the client machine 10 to disconnect.” Ex. 1005 ¶ 581, *cited in* Lin Decl. ¶ 155. We preliminarily determine that these passages adequately show that both Berg and Wookey contemplate a disconnection scenario, and the disclosed systems have ways of handling a disconnection.

On this record, we preliminarily determine these excerpts from Berg and Wookey adequately support Petitioner’s assertion that both Wookey and Berg “aim to maintain connection between nodes through unintentional disconnection periods,” and thus, it would have been obvious to one of ordinary skill in the art to combine them to arrive at the ITP limitation under the disconnection hypothetical. *See* Pet. 56–59. For at least this reason,

IPR2021-00630

Patent 10,075,565 B1

Petitioner has shown sufficiently for purposes of this Decision that the Wookey-Berg combination renders obvious the subject matter recited in the ITP limitation.

Petitioner's disconnection hypothetical (*id.*) also addresses Patent Owner's proposed claim construction for "communicated" and the phrase "causes the second protocol connection to be kept at least partially alive" (Prelim. Resp. 9, 13–24). *See supra* § II.C (claim construction). In particular, Petitioner adequately explains at this stage that if there is a disconnection then packets could not be sent or received. Pet. 56–59. So Patent Owner's arguments about the claim term "communicated" appear to be obviated by the disconnection hypothetical on this record and at this stage. Prelim. Resp. 13–24. Also, the preliminary record shows that, under Petitioner's disconnection hypothetical, there would be no packets communicated to keep the connection active without a connection. Pet. 56–59.

Because Petitioner has sufficiently shown that the ITP limitation would have been obvious under at least one rationale, we do not provide detailed guidance on the other two alternative rationales under Berg's null segment and transfer-state value time periods. *See id.* at 52–55. We, however, encourage the parties to address during trial whether Berg sends ACKs during the time period that Petitioner has identified as the idle time period under Petitioner's two other rationales. Although Patent Owner argues that Berg sends an ACK during the null segment timer-related period, the parties are invited to also address whether Berg sends an ACK during the period specified in the transfer-state timeout-value parameter field. *See* Ex. 1007, 22:25–30.

IPR2021-00630

Patent 10,075,565 B1

On this record, we are sufficiently persuaded by Petitioner's contentions, summarized above, and we disagree with Patent Owner's arguments. In particular, we are persuaded that Petitioner has explained sufficiently how the combination of Wookey and Berg teaches the subject matter recited in claims 28 and that a person of ordinary skill in the art would have had reason to combine the teachings of Wookey and Berg in the manner asserted. Having considered the parties' arguments, we determine that Petitioner has established a reasonable likelihood of prevailing in challenging claim 28 as obvious over the combined teachings of Wookey and Berg.

III. CONCLUSION

For the reasons discussed, we determine that the Petition establishes that there is a reasonable likelihood that Petitioner would prevail in challenging claim 28 of the '565 patent, and we institute *inter partes* review on the ground raised in the Petition. At this stage of the proceeding, we have not made a final determination with respect to the patentability of claim 28 or the construction of any claim term.

IV. ORDER

It is

ORDERED that under 35 U.S.C. § 314(a) and 37 C.F.R. § 42.4, an *inter partes* review is instituted as to claim 28 of the '565 patent on the challenge raised in the Petition; and

FURTHER ORDERED that, under 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is given of the institution of a trial, which will commence on the entry date of this decision.

IPR2021-00630

Patent 10,075,565 B1

FOR PETITIONER:

Naveen Modi

Joseph Palys

Quadeer Ahmed

Jason Heidemann

PAUL HASTINGS LLP

naveenmodi@paulhastings.com

josephpalys@paulhastings.com

quadeerahmed@paulhastings.com

jasonheidemann@paulhastings.com

FOR PATENT OWNER:

Timothy Devlin

DEVLIN LAW FIRM LLC

td-ptab@devlinlawfirm.com

EXHIBIT 17

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner,

v.

JENAM TECH, LLC,
Patent Owner.

IPR2021-00627
Patent 10,375,215 B1

Before DANIEL J. GALLIGAN, SCOTT B. HOWARD, and
JASON M. REPKO, *Administrative Patent Judges*.

HOWARD, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

INTRODUCTION

A. Background and Summary

Google LLC filed a Petition requesting *inter partes* review of claims 1, 4, 8, and 9 of U.S. Patent No. 10,375,215 B1 (Ex. 1001, “the ’215 patent”). Paper 1 (“Petition,” “Pet.”). Jenam Tech, LLC (“Patent Owner”) filed a Preliminary Response. Paper 6 (“Prelim. Resp.”). With our authorization, Petitioner filed a Reply to the Preliminary Response (Paper 8, “Pet. Prelim. Reply”) and Patent Owner filed a Preliminary Sur-Reply (Paper 9, “PO Prelim. Sur-reply”).

We have authority, acting on the designation of the Director, to determine whether to institute an *inter partes* review under 35 U.S.C. § 314 and 37 C.F.R. § 42.4(a). *Inter partes* review may not be instituted unless “the information presented in the petition filed under section 311 and any response filed under section 313 shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a) (2018). “When instituting *inter partes* review, the Board will authorize the review to proceed on all of the challenged claims and on all grounds of unpatentability asserted for each claim.” PTAB Rules of Practice for Instituting on All Challenged Patent Claims and All Grounds and Eliminating the Presumption at Institution Favoring Petitioner as to Testimonial Evidence, 85 Fed. Reg. 79,120, 79,129 (Dec. 9, 2020), (to be codified at 37 C.F.R. 42.108(a)).

B. Real Parties in Interest

Petitioner identifies itself as the real party in interest. Pet. 1. Petitioner further states “Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties-in-interest to this proceeding.” *Id.* at 1 n.1.

IPR2021-00627

Patent 10,375,215 B1

Patent Owner identifies itself as the real party in interest. Paper 4, 1.¹

C. Related Matters

The parties identify the following district court proceedings involving the '215 patent: *Jenam Tech, LLC v. Google, LLC*, No. 6:20-cv-00453 (W.D. Tex.) and *Jenam Tech, LLC v. Samsung Elecs. Co.*, Case No. 4:20-cv-279 (E.D. Tex.) (voluntarily dismissed). Pet. 1; Paper 4, 1.

The parties also identify the following *inter partes* review proceedings: *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00845 (PTAB) (“the ’845 IPR”);² *Unified Patents, LLC v. Jenam Tech, LLC*, IPR2020-00742 (PTAB) (terminated); *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00628 (PTAB); *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00629 (PTAB); and *Google LLC et al. v. Jenam Tech, LLC*, IPR2020-00630 (PTAB). Pet. 2; Paper 4, 1.

D. The ’215 Patent

The ’215 patent states that some transmission control protocol (TCP) implementations support a keep-alive option, but the ’215 patent states that “[e]ach node supports or does not support keep-alive for a TCP connection based on each node’s requirements without consideration for the other node in the TCP connection.” Ex. 1001, 1:54–56, 1:64–67. According to the ’215 patent, “[t]o date no mechanism to allow two TCP connection endpoints to cooperate in supporting the keep-alive option has been proposed or implemented.” *Id.* at 2:25–28. Thus, the ’215 patent states that “there exists

¹ Patent Owner did not add page numbers to its Mandatory Notices. We treat the first page after the cover page as page 1.

² In the ’845 IPR, the Board found that Petitioner proved the challenged claims of US Patent 9,992,995 (“the ’995 patent”) unpatentable. ’845 IPR, Paper 33.

IPR2021-00627

Patent 10,375,215 B1

a need for methods, systems, and computer program products for sharing information for detecting an idle TCP connection.” *Id.* at 2:32–34.

E. Illustrative Claims

Claim 1, the only independent challenged claim, is reproduced below:

1. [1.a] A computer-implemented method,
comprising:

[1.b] at a server node:

causing data to be sent to a client node; and

[1.c] providing access, to the client node, to
code that, in response to being used by the client
node, causes the client node to:

[1.d] receive, by the client node, a
transmission control protocol (TCP)-variant
packet,

[1.e] detect an idle time period
parameter field in the TCP-variant packet,

[1.f] identify metadata in the idle time
period parameter field for an idle time
period, during which, no packet is
communicated in a TCP-variant connection
to keep the TCP-variant connection active,
and

[1.g] determine, based on the
metadata, a timeout attribute associated with
the TCP-variant connection.

Ex. 1001, 24:15–31.

F. Prior Art and Asserted Grounds

Petitioner asserts that claims 1, 4, 8, and 9 would have been unpatentable on the following grounds:

IPR2021-00627

Patent 10,375,215 B1

Claim(s) Challenged	35 U.S.C. § ³	Reference(s)/Basis
1	103	Wookey ⁴ , Eggert ⁵
4, 8, 9	103	Wookey, Eggert, Abdolbaghian ⁶

Petitioner also relies on the declaration testimony of Dr. Bill Lin (Ex. 1002).

ANALYSIS

A. Legal Standards

In *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1 (1966), the Supreme Court set out a framework for assessing obviousness under 35 U.S.C. § 103 that requires consideration of four factors: (1) the “level of ordinary skill in the pertinent art,” (2) the “scope and content of the prior art,” (3) the “differences between the prior art and the claims at issue,” and (4) if in evidence, “secondary considerations” of non-obviousness such as “commercial success, long-felt but unsolved needs, failure of others, etc.” *Id.* at 17–18. “While the sequence of these questions might be reordered in any particular case,” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 407 (2007), the U.S. Court of Appeals for the Federal Circuit has repeatedly

³ The Leahy-Smith America Invents Act (“AIA”) included revisions to 35 U.S.C. §§ 102 and 103 that became effective March 16, 2013. The application for the ’215 patent was filed on July 19, 2018. Although the ’215 patent claims priority to applications filed before March 16, 2013, Patent Owner has not shown that the written description of the earlier applications supports the challenged claims. *See Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378–80 (Fed. Cir. 2015). Therefore, we apply the post-AIA versions of 35 U.S.C. §§ 102 and 103.

⁴ US 2007/0171921 A1, published July 26, 2007 (Ex. 1005).

⁵ L. Eggert, “TCP Abort Timeout Option,” Apr. 14, 2004 (Ex. 1006).

⁶ US 6,981,048 B1, issued Dec. 27, 2005 (Ex. 1007).

IPR2021-00627

Patent 10,375,215 B1

emphasized that “it is error to reach a conclusion of obviousness until all those factors are considered,” *WBIP v. Kohler*, 829 F.3d 1317, 1328 (Fed. Cir. 2016).

B. Level of Ordinary Skill in the Art

In determining whether an invention would have been obvious at the time it was made, we consider the level of ordinary skill in the pertinent art at the time of the invention. *Graham*, 383 U.S. at 17. “The importance of resolving the level of ordinary skill in the art lies in the necessity of maintaining objectivity in the obviousness inquiry.” *Ryko Mfg. Co. v. Nu-Star, Inc.*, 950 F.2d 714, 718 (Fed. Cir. 1991). The “person having ordinary skill in the art” is a hypothetical construct, from whose vantage point obviousness is assessed. *In re Rouffet*, 149 F.3d 1350, 1357 (Fed. Cir. 1998).

Factors pertinent to a determination of the level of ordinary skill in the art include “(1) the educational level of the inventor; (2) type of problems encountered in the art; (3) prior art solutions to those problems; (4) rapidity with which innovations are made; (5) sophistication of the technology; and (6) educational level of active workers in the field.” *Envtl. Designs, Ltd. v. Union Oil Co. of Cal.*, 713 F.2d 693, 696–697 (Fed. Cir. 1983) (citing *Orthopedic Equip. Co. v. All Orthopedic Appliances, Inc.*, 707 F.2d 1376, 1381–82 (Fed. Cir. 1983)). “Not all such factors may be present in every case, and one or more of these or other factors may predominate in a particular case.” *Id.*

Petitioner argues that a person having ordinary skill in the art “would have an undergraduate degree in electrical engineering, computer engineering, computer science or a related field along with at least two years of work experience in the field of networking.” Pet. 5 (citing Ex. 1002

IPR2021-00627

Patent 10,375,215 B1

¶¶ 16–18). Petitioner further argues that “[m]ore education can supplement practical experience and vice versa.” *Id.* (citing Ex. 1002 ¶¶ 16–18).

Patent Owner substantially agrees with Petitioner’s proposal. Prelim. Resp. 5.

Accordingly, for purposes of this Decision, we adopt Petitioner’s proposed formulation of the level of ordinary skill in the art, except that we delete the qualifier “at least” to eliminate vagueness as to the amount of practical experience. The qualifier expands the range indefinitely without an upper bound, and thus precludes a meaningful indication of the level of ordinary skill in the art.

C. Claim Construction

We apply the same claim construction standard used in the federal courts, in other words, the claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. § 282(b), which is articulated in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). *See* 37 C.F.R. § 42.100(b) (2020). Under the *Phillips* standard, the “words of a claim ‘are generally given their ordinary and customary meaning,’” which is “the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application.” *Phillips*, 415 F.3d at 1312–13.

Petitioner argues that no terms need to be construed. Pet. 10–11.

Patent Owner proposes a construction of the phrase “idle time period, during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active.” Prelim. Resp. 7–23. This is sometimes referred to as the “ITP.” *See id.* at 7. Specifically, Patent Owner argues that the ITP means “idle time period and, during which, no packet of

IPR2021-00627

Patent 10,375,215 B1

any kind is sent or received by the apparatus in the TCP-variant connection to keep the TCP-variant connection active with or without a condition and via any mechanism.” *Id.*

1. *Idle Time Period*

Claim 1 recites “identifying metadata in the idle time period parameter field for an *idle time period, during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active.*” Ex. 1001, 24:25–29 (emphasis added).

Patent Owner’s proposed constructions for the claim recitations italicized above are in the following table:

Claim Language	Patent Owner’s Proposed Construction
“idle time period and, during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active” (Claim 1)	“ <i>idle time period</i> and, during which, no packet <i>of any kind</i> is <i>sent or received by the apparatus</i> in the TCP-variant connection to keep the TCP-variant connection active <i>with or without a condition and via any mechanism</i> ”

Prelim. Resp. 7; *see also id.* at 7–23 (argument concerning these constructions).

As set forth in the table above, Patent Owner’s proposed construction is based on the following three terms or phrases in the recited subject matter: “no packet,” “communicated,” and “to keep the TCP-variant connection active.” Prelim. Resp. 8–23. First, as to the language “no packet,” Patent Owner argues that “[t]here are no qualifiers nor adjectives recited in connection with the term ‘packet’ and none should be inserted through erroneous construction or interpretation.” *Id.* at 9. Based on the current record, we disagree. Although the phrase “no packet” is absolute, the

IPR2021-00627

Patent 10,375,215 B1

remainder of the claim includes a significant qualifier by reciting that no packet is communicated “to keep the TCP-variant connection active.”

Ex. 1001, 24:25–29. Thus, we have no issue with interpreting the words “no packet” in isolation to mean “no packet of any kind,” but we cannot ignore the phrase “to keep the TCP-variant connection active.” Thus, the phrase “no packet of any kind” is still qualified by other language in the claim.

Second, Patent Owner argues that the term “communicated” means “sent or received by the apparatus.” Prelim. Resp. 12–21. According to Patent Owner, “since the claims refer to the connection in which the system is a part, ‘communicated’ refers to packets being sent or received by that system.” *Id.* at 13. Stated differently, Patent Owner argues that “‘communicated’ should not be limited to **only** [‘]receiving,’ ‘sending,’ or ‘receiving **and** sending.’ It is broader. Encompassing sending **or** receiving is consistent with the claim’s scope and claim differentiation.” *Id.* at 15.

As explained below, we determine on this record that Petitioner has shown that the asserted prior art teaches “communicated” under Patent Owner’s proposed construction. Therefore, for purposes of this Decision, we apply Patent Owner’s construction of “communicated,” and we need not resolve any disputes between the parties over the scope of this term. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (“[W]e need only construe terms ‘that are in controversy, and only to the extent necessary to resolve the controversy’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

Third, Patent Owner argues that the phrase “to keep the TCP-variant connection active” means “to keep the TCP-variant connection active with or without a condition and via any mechanism.” Prelim. Resp. 7, 22–23.

IPR2021-00627

Patent 10,375,215 B1

As explained below, we find that the asserted prior art teaches, in at least one scenario, an absence of packets that are communicated to keep the connection active, and, therefore, we apply Patent Owner’s construction of the phrase “to keep the TCP-variant connection active” and need not resolve any disputes regarding the scope of this phrase.

2. *Remaining Terms*

For purposes of this Decision, we need not construe expressly any other claim terms. *See Nidec*, 868 F.3d at 1017 (noting that “we need only construe terms ‘that are in controversy, and only to the extent necessary to resolve the controversy’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

D. *Discretionary Denial under Section 314(a)*⁷

Patent Owner argues that we should exercise discretion to deny the Petition under 35 U.S.C. § 314(a) based on the factors set forth in *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 (PTAB Sept. 6, 2017) (designated precedential in relevant part). Prelim. Resp. 57–69. Patent Owner cites other petitions for *inter partes* review that have been filed challenging claims of Patent Owner’s patents and argues that, “[b]ecause [Petitioner’s] petitions challenge related patents that cover the same subject matter and include overlapping claim limitations, these follow-on petitions place an unfair burden on Patent Owner and should be denied.” Prelim. Resp. 59.

⁷ Although the Preliminary Response identifies both sections 314(a) and 325(d) in the heading, the body of the argument only discusses section 314(a). *See* Prelim. Resp. 57–68. Accordingly, we only address section 314(a) here.

IPR2021-00627

Patent 10,375,215 B1

Under 35 U.S.C. § 314(a), the Director has discretion to deny institution of an *inter partes* review. *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2140 (2016) (“[T]he agency’s decision to deny a petition is a matter committed to the Patent Office’s discretion.”). Relying on this discretionary authority, Patent Owner argues that the Board should deny institution because Petitioner has, *inter alia*, already filed a petition challenging a related patent in the ’845 IPR. Prelim. Resp. 59–62 (citing *Gen. Plastic Indus. Co. v. Cannon Kabushiki Kaisha*, IPR2016-01357, Paper 19 at 9–10 (PTAB Sept. 6, 2017) (Section II.B.4.i precedential); *Tietex Int’l, Ltd. v. Precision Fabrics Group, Inc.*, IPR2015-01671, Paper 7, 8 (PTAB Feb. 11, 2016)); *see also id.* at 62–64 (discussing the ’845 IPR).

In *General Plastic*, the Board articulated a non-exhaustive list of factors to be considered in determining whether to exercise discretion under § 314(a) to deny a petition that challenges the same patent as a previous petition. *Gen. Plastic* at 9–10. The *General Plastic* factors follow:

1. whether the same petitioner previously filed a petition directed to the same claims of the same patent;
2. whether at the time of filing of the first petition the petitioner knew of the prior art asserted in the second petition or should have known of it;
3. whether at the time of filing of the second petition the petitioner already received the patent owner’s preliminary response to the first petition or received the Board’s decision on whether to institute review in the first petition;
4. the length of time that elapsed between the time the petitioner learned of the prior art asserted in the second petition and the filing of the second petition;
5. whether the petitioner provides adequate explanation for the time elapsed between the filings of multiple petitions directed to the same claims of the same patent;

IPR2021-00627

Patent 10,375,215 B1

6. the finite resources of the Board; and

7. the requirement under 35 U.S.C. § 316(a)(11) to issue a final determination not later than 1 year after the date on which the Director notices institution of review.

Id. at 9–10.

As with other non-dispositive factors considered for institution under § 314(a), the Board weighs factors as part of a “balanced assessment of all relevant circumstances in the case, including the merits.” Patent Trial and Appeal Board Consolidated Trial Practice Guide 58 & n.2 (Nov. 2019) (“CTPG”)⁸ (discussing follow-on petitions and parallel proceedings, citing *General Plastic*).

We do not agree with Patent Owner’s argument that this is the type of “follow-on” petition that should be denied under the Board’s precedential *General Plastic* decision. *General Plastic* sets forth a series of factors to be considered by the Board in evaluating whether to exercise discretion under § 314(a) to deny a petition that challenges a patent that was previously challenged before the Board. *General Plastic*, Paper 19 at 15–19. As *General Plastic* states, “[m]ultiple, staggered petitions *challenging the same patent and same claims* raise the potential for abuse.” *Id.* at 17 (emphasis added). Here, Patent Owner cites numerous petitions against related patents but none of the other petitions challenge claims of the ’215 patent. Prelim. Resp. 58–59. Thus, this is not a situation involving “[m]ultiple, staggered petitions challenging the same patent and same claims” contemplated by *General Plastic*. See *General Plastic*, Paper 19 at 17.

⁸ Available at <https://www.uspto.gov/TrialPracticeGuideConsolidated>.

IPR2021-00627

Patent 10,375,215 B1

Patent Owner's reliance on *Tietex Int'l, Ltd. v. Precision Fabrics Group, Inc.*, IPR2015-01671, Paper 7, 8 (PTAB Feb. 11, 2016)) is misplaced. *See* Prelim. Resp. 57, 62, 64. In that proceeding, the petitioner requested *inter partes* review of claims of a patent based on three prior art references. *Tietex*, Paper 7 at 1, 4. In a prior proceeding involving a related patent, the Board entered a Final Written Decision finding that the petitioner had not shown the claims of that related patent were unpatentable based on the same prior art. *Id.* at 7. The Board compared the arguments of the two proceedings and determined that in the second proceeding, the petitioner had not "offered any argument or evidence that significantly differs from that offered in the petition in the [earlier] IPR that is likely to lead to a different result and, thus, would justify instituting an *inter partes* review in this proceeding." *Id.* at 9. Because the Board had previously found the arguments raised by petitioner in the petition not sufficient, the Board determined that "that instituting an *inter partes* review on Petitioner's asserted ground . . . would not be an efficient use of administrative resources." *Id.*

Tietex is distinguishable from the facts of this case. In *Tietex*, the Board relied on a Final Written Decision in an earlier case on a related patent finding that the petitioner did not prove the challenged claims unpatentable. *See Tietex*, Paper 7 at 9. In other words, because the Board previously found the petitioner's arguments unpersuasive after a trial, the same evidence and arguments were likely to be unpersuasive in the second case and instituting trial would not have been an efficient use of resources. Here, in contrast, the Final Written Decision in the earlier case involving the related '995 patent found the challenged claims unpatentable. *Google LLC v. Jenam Tech, LLC*, IPR2020-00845, Paper 33 at 39 (PTAB Aug. 27,

IPR2021-00627

Patent 10,375,215 B1

2021). Thus, unlike in *Teitex*, because Petitioner succeeded in the earlier case, those same arguments are likely to be found sufficient after trial in this case.⁹ Thus, the facts of this case stand in sharp contrast to those in *Tietex*.

Accordingly, we decline to exercise our discretion to deny institution under section 314(a).

E. Asserted Obviousness in View of Wookey and Eggert

Petitioner argues that claim 1 would have been obvious over Wookey and Eggert. *See* Pet. 12–52. Based on the current record, we are persuaded that Petitioner has established a reasonable likelihood of prevailing on this asserted obviousness ground with respect to claim 1.

1. Wookey

Wookey is generally directed “providing access to computing environments” and, more specifically, “to methods and systems for making a hypermedium page interactive.” Ex. 1005 ¶ 2.

⁹ This is not to say Patent Owner is foreclosed to presenting arguments in this proceeding. To the contrary, unless issue preclusion applies, we will consider all arguments made in this case *de novo* and will reach our determination based on the record in this case, not the record in the earlier proceeding.

IPR2021-00627

Patent 10,375,215 B1

Wookey Figure 1 is reproduced below.

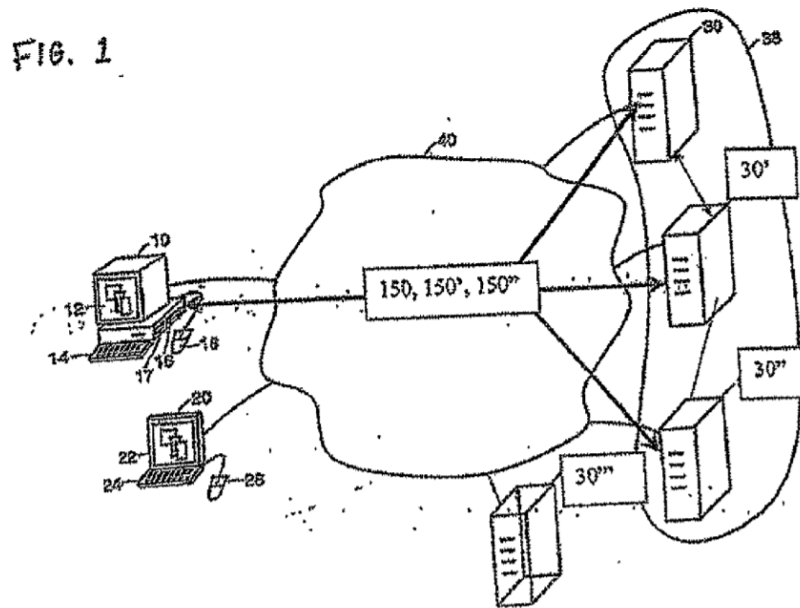


Figure 1 “is a block diagram of one embodiment of an environment in which a client machine accesses a computing resource provided by a remote machine.” Ex. 1005 ¶ 20. Figure 1 shows “a client machine 10, 10’ accesses a computing resource provided by a remote machine, 30, 30’, 30”, 30”.” *Id.* ¶ 135.

2. Eggert

Eggert “specifies a new TCP option - the Abort Timeout Option - that allows conforming hosts to negotiate per-connection abort timeouts. This allows mobile hosts to maintain TCP connections across disconnected periods that are longer than their system’s default abort timeout.” Ex. 1006, 3.

3. Analysis of Claim 1

a) Undisputed Limitations

Claim 1 recites “[a] computer-implemented method,” “at a server node,” “causing data to be sent to a client node,” and a client node which

“receive[s], by the client node, a transmission control protocol (TCP)-variant packet,” and “determine[s], based on the metadata, a timeout attribute associated with the TCP-variant connection.” Ex. 1001, 24:15–31. This corresponds to limitations [1a], [1b], [1d], and [1g]. Petitioner contends that the combination of Wookey and Eggert teaches each of those limitations. Pet. 12–18, 22–43, and 50–52 (citing Ex. 1002; Ex. 1005; Ex. 1006).

After reviewing Petitioner’s arguments and information regarding the limitations identified above, including the Lin Declaration, which are not addressed by Patent Owner at this stage (*see* Prelim. Resp.), we are persuaded that Petitioner sufficiently demonstrates, for purposes of this Decision, that the combination of Wookey and Eggert teaches each of those limitations.

b) Providing Access, to the Client Node, to Code That, in Response to Being Used by the Client Node, Causes the Client Node to”—Limitation [1c]

Claim 1 further recites “providing access, to the client node, to code that, in response to being used by the client node, causes the client node to” perform a list of enumerated steps. Ex. 1001, 24:18–20. We refer to this limitation as “the providing step.”

Petitioner argues that Wookey teaches the providing step. Pet. 18–22. Specifically, Petitioner argues that either (1) the encoded Uniform Resource Locator (“URL”) associated with icons on an HTML page 288 or (2) HTML page 288 teaches the code limitation recited in claim 1. *Id.* at 19, 19 n.12. We address the second argument—HTML page 288 as code—below.¹⁰

¹⁰ Because we find, for purposes of this Decision, that Petitioner has sufficiently shown that HTML page 288 is the recited code, we need not address whether a URL address is code.

IPR2021-00627

Patent 10,375,215 B1

Petitioner argues that Wookey teaches that “remote machine 30 provides access, to client machine 10, to encoded Uniform Resource Locators (URLs) (‘code’) associated with icons on an HTML page 288.” Pet. 19 (citing Ex. 1002 ¶¶ 119–120).¹¹ Petitioner further argues that “HTML page 288 (including the encoded URLs) also discloses the claimed ‘code’ because an HTML page was known to comprise HTML code used to define, e.g., what is displayed on the screen of the client node.” *Id.* at 22 (citing Ex. 1002 ¶ 124). Petitioner further argues that that “[t]he HTML code used to display information relating to page 288 would include the icons associated with the URLs that the client node uses to access a desired resource, as further discussed below for limitation 1.d” and “HTML page 288 includes code.” *Id.* (citing Ex. 1002 ¶¶ 124–125).

Petitioner further argues that “[t]his understanding (e.g., HTML page includes code) is consistent with PO’s interpretation of this limitation in its infringement positions for the ’215 patent. (EX1017, 2 (‘code (e.g., also in the HTML pages, etc.)’).)” *Id.*

Although Patent Owner extensively argues that Wookey’s encoded URL does not teach the recited “providing step” (Petitioner’s other argument), Patent Owner does not address Petitioner’s argument based on HTML page 288 being “code.” *See* Prelim. Resp. 23–45.

Based on the current record, we are sufficiently persuaded by Petitioner’s argument that Wookey’s HTML page 288 is code and that Wookey teaches the providing limitation set forth in claim 1. Specifically,

¹¹ The parties have italicized the names of the prior art references and Petitioner has colorized certain words in the prior art. Those italics and coloration have been omitted from this Decision.

Dr. Lin’s testimony provides the basis for his opinion that an HTML page is code. *See* Ex. 1002 ¶ 124 (citing Ex. 1020 ¶¶ 20, 24, 26, 79); *see also id.* ¶¶ 61–62 (describing HTML). Additionally, Dr. Lin’s opinion is consistent with Patent Owner’s position in the district court limitation. *See* Ex. 1017, 2 (Patent Owner’s First Set of Infringement Contentions) (“providing access, to the client node, to *code (e.g., also in the HTML pages, etc.)* that, in response to being used by the client node, causes the client node to carry out the following functionality.” (emphasis added)). Petitioner further sets forth how the code is provided to the client server and why Wookey’s HTML page 288 causes the client node to perform the steps set forth in the remainder of claim 1. *See* Ex. 1002 ¶¶ 119–120, 125.

c) The ITP Limitation – Limitations [1e] and [1f]

Two operations recited in claim 1 that the code causes the client node to perform are “detect an idle time period parameter field in the TCP-variant packet” and “identify metadata in the idle time period parameter field for an idle time period, during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active.” Ex. 1001, 24:23–29.

(1) Petitioner’s Arguments

Petitioner argues that Eggert teaches detecting an abort timeout field and identifying metadata in that field (the value of the timeout) in the three-way handshake of Figure 2. Pet. 41–46 (discussing exchange of ATO values in Eggert’s Figure 2). According to Petitioner, the abort timeout value contained in the ATO field that is exchanged during Eggert’s three-way handshake teaches “an idle time period parameter field,” as recited in claim 1. Pet. 41–43 (citing Ex. 1002 ¶¶ 172–181; Ex. 1006, 1–7, Fig. 1). Petitioner also argues that Eggert’s description of the “abort timeout value defines an idle time period because it is associated with a period where no

IPR2021-00627

Patent 10,375,215 B1

packet is communicated in the connection to keep the connection active.” Pet. 43 (citing Ex. 1002 ¶¶ 177–179; Ex. 1006, Abstract); *see also id.* at 45 (“Eggert further discloses that the timeout value in the ATO field corresponds to an idle time period, ‘during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active,’ as claimed.” *Id.* (citing Ex. 1002 ¶ 187)).

Petitioner further argues that an acknowledge message (“ACK”) sent in response to a packet is a packet that is communicated to keep the connection active. Pet. 46. Specifically, Petitioner argues Eggert relies on “[t]he TCP specification [1] [which] includes a ‘user timeout’ that defines the maximum amount of time that segments may remain *unacknowledged* before TCP will abort the connection. If a disconnection lasts longer than the user timeout, the TCP connection will abort.” *Id.* (quoting Ex. 1006, 3) (alteration and emphasis added in Petition). Thus, according to Petitioner, when no ACK is received within a certain period of time after sending a packet, the connection will end. *See id.*

Petitioner also focuses on what happens in the Wookey/Eggert combination when there is a disconnection between client machine 10 and remote machine 30. *See* Pet. 46–49. This is sometimes referred to as the “disconnection hypothetical” or “Disconnect-Hypo.” Specifically, Petitioner argues that “when an interruption in communication occurs such as a disconnection in the network service (as contemplated by Wookey and Eggert) or in a physical network medium (as described in the ’215 patent) (EX1001, 2:11–14, 2:28–31, 12:67–13:6), packets from one node do not reach the other node in either direction.” *Id.* at 47 (citing Ex. 1002 ¶ 192). According to Petitioner, “[i]n this situation contemplated by the ’215 patent, no packets are communicated between the nodes to keep the connection

IPR2021-00627

Patent 10,375,215 B1

active because they cannot get through the interruption or the physical medium's disconnection.” *Id.* (citing Ex. 1002 ¶ 192; Ex. 1001, 12:64–13:6, 2:28–31, 22:22–30, Fig. 7).

(2) *Patent Owner's Arguments*

Patent Owner argues that Petitioner has changed its hypothetical from the one used in the '845 IPR and ignored its prior admissions. Prelim. Resp. 46–47. Patent Owner further argues that the disconnection hypothetical is just that, “hypothetical” and not taught in Eggert. *Id.* at 47–49.

Patent Owner further argues that even if the disconnection hypothetical took place, it would not invoke Eggert's ATO feature. Prelim. Resp. 49–51. Specifically, Patent Owner argues that “Eggert requires an ACK-eliciting data packet to be sent before the ATO timer is triggered” but that “[n]o such ACK-eliciting data packet is shown to be sent in Petitioners new, undisclosed *Disconnect-Hypo* shown above.” *Id.* at 49–50. According to Patent Owner, “[b]ecause there is no ACK-eliciting data packet and the ATO timer could thus not even start, Google's *Disconnect-Hypo* is unworkable—the ATO **cannot** trigger a closure.” *Id.* at 50 (emphasis in original).

Additionally, Patent Owner argues that ACK packets are not “the only packets capable of keeping Eggert's connection active.” Prelim Resp. 51. Patent Owner argues Petitioner's expert admitted in a different proceeding that “retransmitted-packets **are** sent to keep the connection active.” *Id.*¹²

¹² As support, Patent Owner directs as to exhibits filed in a different proceeding. However, “[a]ll evidence must be filed in the form of an exhibit.” 37 C.F.R. § 42.63(a). If a party wishes us to consider papers or exhibits from a different proceeding, the party must file them as an exhibit in this proceeding.

IPR2021-00627

Patent 10,375,215 B1

More specifically, Patent Owner argues that Petitioner's expert previously testified that

In the example below, device 1 is transmitting a packet of data to device 2. The packet of data gets lost before it reaches device 2. ***Device 1 has a locally defined retransmission timeout D1.*** Timeout D1 is a logically set up retransmission timer of when device 1 is to expect the acknowledgment from device 2. ***When this retransmission timer runs out and no acknowledgment is received, the packet of data is retransmitted by device 1.***

Id. at 52. Based on those admissions, Patent Owner argues that in the disconnection hypothetical,

the retransmission timer will inevitably expire in order to retransmit during the ATO period to maintain the connection. That is, Google defined the “conditions” of their *Disconnect-Hypo* and admitted the ***same*** “conditions” cause retransmitted-packets. Taken together, Google concedes that retransmitted-packets ***are*** sent in its *Disconnect-Hypo* to keep the connection active.

Id. at 53.

Patent Owner argues that is confirmed by RFC 793, upon which Eggert is based. Prelim Resp. 53 (citing Ex. 1006, 8, 14).¹³

(3) *Our Analysis*

For the reasons explained below and based on the current record, we are sufficiently persuaded that Eggert's ATO field is an idle time parameter

¹³ Patent Owner also addresses argument that Petitioner made in the '845 IPR. *See* Prelim. Resp. 54–57. To the extent Patent Owner is attempting to argue the '845 IPR after the close of briefing, that is improper. To the extent that Patent Owner addresses arguments that have not been made in this proceeding, such arguments are not yet ripe and will not be addressed unless Petitioner makes the same arguments and Patent Owner responds during the trial.

IPR2021-00627

Patent 10,375,215 B1

field for an abort timeout (or idle time period) which will abort the connection if, during the period, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection alive. Our determination is based on what happens in Eggert—along with the combination of Eggert and Wookey—if there is a disconnection after an ATO period is negotiated in the three-way handshake.

Based on the current record, Eggert’s TCP-variant protocol uses the three-way handshake to establish a connection between two nodes on a network. Ex. 1006, 5; Ex. 1002 ¶ 149. The three-way handshake involves the exchange of synchronize (SYN) and ACK messages between the nodes to create a connection—i.e., SYN, SYN-ACK, and ACK message in a standard TCP implementation. Ex. 1002 ¶ 149; Ex. 1008, 31–32, 34–37.¹⁴ During the three-way handshake, the client node and the server node agree upon abort timeout (ATO) period which is longer than the user timeout period set in the TCP specification. Ex. 1006, 3. “This allows mobile hosts to maintain TCP connections across disconnected periods that are longer than their system’s default abort timeout.” *Id.* As part of that communication, the client node will receive a packet with the ATO parameter field and will then identify the metadata in that field establishing the ATO period. *Id.* at 5–7. That ATO period negotiated in Eggert sets the time period that must pass without an ACK before the connection is terminated. *See* Ex. 1002 ¶¶ 83–85, 138, 142, 144.

¹⁴ It appears that the citations are to the native pagination and not the page numbers added to the exhibits. Petitioner should clarify if that is not correct. Our analysis will refer to the exhibit page number and not the native page number.

Based on the current record, a disconnection—such a disconnection in the network service or a physical network medium—between a client node and a server node using a Wookey/Eggert connection would cause the TCP-variant connection to abort the connection. Specifically, because of the disconnection, no packets, including ACK packets, can get through the interruption. *See* Ex. 1002 ¶ 192. For example, if a packet is sent by the client node after the disconnection happens (*see* Pet. 47–49), the server node will not receive the packet and cannot send an ACK packet. Ex. 1002 ¶ 192. Once the original packet is sent and the ATO timer started, and no ACK will be sent or received, and, as a result, once the ATO timer has expired, the connection will be aborted. *See id.*; Ex. 1006, 3; Ex. 1012, 3:5–25; Ex. 1008, 77. In other words, based on the current record, during the disconnection hypothetical, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active.

The retransmission timer cited by Patent Owner is a separate timer that governs retransmission of data if an ACK is not received within a particular timeout interval. *See* Prelim. Resp. 53 (citing Ex. 1008, 8, 14). RFC 793—which is the Transmission Control Protocol Specification published by the DARPA internet program—explains that, “[w]hen the TCP transmits a segment containing data, it puts a copy on a retransmission queue and starts a timer; when the acknowledgment for that data is received, the segment is deleted from the queue.” Ex. 1008, 14. However, “[i]f the acknowledgment is not received before the timer runs out, the segment is retransmitted.” Ex. 1008, 14. There is no discussion of disconnecting the connection if a retransmission message is not sent. *Id.* Based on the current record, RFC 793 and Eggert do not teach, and Patent Owner does not argue, that retransmitting data resets the original timeout timer that started upon

IPR2021-00627

Patent 10,375,215 B1

sending the data the first time. Thus, despite retransmitting the data, the connection will terminate on the expiration of the ATO timer that started when the first data were transmitted.

Patent Owner argues that the language “to keep the TCP-variant connection active” describes a “purpose” of the packet. *See* Prelim. Resp. 55. If the purpose of retransmitting the data is to keep the connection active, then the device would be programmed to reset the ATO timer upon retransmission to avoid termination of the connection. The fact that this is not done supports a determination, based on the current record, that the purpose of these packets is not to keep the connection active. Instead, the purpose of the retransmission message is an attempt to retransmit data that was not properly received and acknowledged. For all of these reasons, based on the current record, we determine that retransmission packets in Eggert are not communicated “to keep the TCP-variant connection active

Accordingly, based on the current record and for the reasons given above, we are sufficiently persuaded by Petitioner’s contention that, under at least certain operating conditions, Eggert teaches the recited ITP subject matter, which is sufficient to show obviousness. *See Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016) (“[C]ombinations of prior art that sometimes meet the claim elements are sufficient to show obviousness.”); *Hewlett–Packard Co. v. Mustek Sys., Inc.*, 340 F.3d 1314, 1326 (Fed. Cir. 2003) (“[A] prior art product that sometimes, but not always, embodies a claimed method nonetheless teaches that aspect of the invention.”).

d) Reason to Combine

Petitioner argues that a person having ordinary skill in the art would have “configure[d] and implement[ed] Wookey’s client machine 10 to use a

IPR2021-00627

Patent 10,375,215 B1

TCP-variant protocol like the one disclosed by Eggert when establishing (and using) the second connection between client machine 10 and remote machine 30’ concerning claim limitations 1.d–1.g.” Pet. 33 (citing Ex. 1002 ¶ 157). According to Petitioner, “[s]uch an implementation would have had many advantages and been a predictable and straightforward result achieved by combining well-known technologies using known methods.” *Id.* (citing Ex. 1002 ¶ 157; *KSR*, 550 U.S. at 416–18).

Specifically, Petitioner argues that “Eggert and Wookey disclose features in a similar technological field.” Pet. 33 (citing Ex. 1002 ¶ 158). According to Petitioner, both references “are directed to establishing reliable connections between two nodes, where the nodes may be mobile nodes that experience an unintentional disconnection, so a POSITA would have had reason to consider Eggert when contemplating and implementing Wookey’s teachings.” *Id.* at 34 (citing Ex. 1002 ¶¶ 159–161).

Petitioner further argues that Wookey teaches “establish[ing] a connection between client machine 10 and remote machine 30’ using a protocol that can be different from the one used to connect client machine 10 and remote machine 30.” Pet. 35 (citing Ex. 1002 ¶¶ 162–163). Petitioner also argues that a person having ordinary skill in the art would have been “motivated by Wookey’s disclosures and suggestions to consider protocols that would provide these desired features for the connection with remote machine 30’, including the features associated with the TCP-variant connection described by Eggert” and, as a result, “would have been motivated to incorporate a protocol like *Eggert*’s TCP-variant protocol in the use of the ‘code’ by client machine 10 for establishing a connection with remote machine 30’.” *Id.* (citing Ex. 1002 ¶¶ 162–163).

IPR2021-00627

Patent 10,375,215 B1

Petitioner further argues that “configuring Wookey to utilize a TCP-variant protocol based on Eggert’s teachings would have been both a predictable and straightforward implementation.” Pet. 36 (citing Ex. 1002 ¶ 165; Ex. 1005 ¶ 216; *KSR*, 550 U.S. at 416); *see also id.* at 37–38 (discussing use of known technologies to achieve a foreseeable result).

Patent Owner argues Petitioner improperly combined embodiments from Wookey. *See* Prelim. Resp. 26–27, 31–36. Specifically, Patent Owner argues that Petitioner improperly combines a discussion of a Figure 3 embodiment for the discussion of code with a Figure 25 embodiment for a teaching of using a different connection protocol. *Id.*

Based on the current record, Petitioner has sufficiently shown a reason to combine the relevant teachings of Wookey and Eggert. Both Wookey and Eggert are directed to similar technology—maintaining connections when a node loses connectivity. *See* Ex. 1005 ¶¶ 581 (discussing how a disconnection can happen), 1135–1136 (discussing the need for connections to manage unintentional disconnection between the client and remote machines); Ex. 1006, 1–3 (discussing how “[l]engthening abort timeouts allows established TCP connections to survive periods of disconnection” as “[s]ome hosts are only intermittently connected to the Internet”). Because both Wookey and Eggert discuss establishing reliable connections between two nodes, where the nodes may be mobile nodes that experience an unintentional disconnection, we are sufficiently persuaded that a person having ordinary skill in the art would have had reason to implement Eggert’s abort timeout negotiation into Wookey’s teachings. *See* Ex. 1002 ¶¶ 159–163; *see also id.* at ¶¶ 168–169 (discussing how a person having ordinary skill in the art would have been able to implement Eggert’s system into Wookey).

IPR2021-00627

Patent 10,375,215 B1

Furthermore, we find that Petitioner has sufficiently shown that Wookey teaches using a different connection protocol between client machine 10 and remote machine 30' than was used initially between client machine 10 and server node/remote machine 30. *See* Ex. 1005 ¶¶ 731–732. Specifically, Wookey teaches that although client machine 10 and remote machine 30 may use a TCP/IP protocol, it may use the IPX protocol to connect with remote machine 30'. *Id.* ¶ 732.

We do not agree with Patent Owner's argument that Petitioner is improperly combining two distinct embodiments from Wookey—the Figure 3D embodiment for the code and the Figure 25 embodiment for the different type of connection—to support the reason to combine. Instead, based on the current record, Figure 25, like Figure 3D, is simply describing how a client machine 10 interacts with a remote machine 30 to access a resource, such as remote machine 10. That is, Wookey teaches different aspects of communication within the environment shown in Figure 1 in the various figures. *See* Ex. 1005 ¶ 135 (discussing environment containing client machines 10 and remote machines 30). Figure 3D focuses on how the “client machine can access a resource from a resource neighborhood web page displayed at that client machine” while Figure 25 depicts “a remote machine and client machine establishing a protocol stack from communication.” *Id.* ¶¶ 26, 57.

Moreover, even if they are different embodiments, Patent Owner overstates Petitioner's reliance on Figure 25. Petitioner is not relying on the specific communication features set forth in the specification. Rather, Petitioner is simply relying on Wookey paragraph 732 for the broad teaching that different communication protocols can be used. *See* Pet. 23, 26. Based on the current record, we determine that Petitioner's reliance on Wookey's

IPR2021-00627

Patent 10,375,215 B1

teachings of the use of different protocols in paragraph 732 is appropriate to support the rationale to combine.

e) Conclusion Regarding Claim 1

Based on the current record, there is a reasonable likelihood that Petitioner will be prove the claim 1 is unpatentable over Wookey and Eggert.

F. Asserted Obviousness in View of Wookey, Eggert, and Abdolbaghian

Petitioner argues that claim 4, 8, and 9 would have been obvious over Wookey, Eggert, and Abdolbaghian. *See* Pet. 52–71. Based on the current record, we are persuaded that Petitioner has established a reasonable likelihood of prevailing on this asserted obviousness ground with respect to claim 1.

1. Abdolbaghian

Abdolbaghian “is generally directed to a system and method in which a second application (the ‘keep-alive function’) sends a ping, message or other signal to a first application to prevent the first application from being timed out (hereinafter, ‘keep-alive input’).” Ex. 1007, code (57). “In an embodiment of the invention, the first application is queried as to the appropriate content and format of a timeout period restart message. The keep-alive function may be automatically executed upon the occurrence of a triggering event.” *Id.*

2. Claims 4, 8, and 9

Because Petitioner has demonstrated a reasonable likelihood of success in proving that at least one claim of the ’215 patent is unpatentable, we institute on all grounds and all claims raised in the Petition. *See* 37 C.F.R. § 42.108(a). Therefore, at this stage of the proceeding, it is not necessary for us to provide an assessment of every challenge raised by

IPR2021-00627

Patent 10,375,215 B1

Petitioner, especially as Patent Owner has not presented any responsive argument regarding the dependent claims.

Nevertheless, we note that Petitioner provides detailed explanations supported by the testimony of Dr. Lin and specific citations to the relevant references indicating where in the references Petitioner argues the limitations of claims 4, 8, and 9 are taught and why a person of ordinary skill in the art would have combined the various teachings of Wookey, Eggert, and Abdolbaghian. *See* Pet. 52–71. Accordingly, at this stage of the proceeding, we are persuaded the information presented in the Petition establishes there is a reasonable likelihood that Petitioner would prevail on its assertion that claims 4, 8, and 9 are unpatentable over Wookey, Eggert, and Abdolbaghian.

G. Arthrex

Relying on *United States v. Arthrex, Inc.*, Nos. 19-1434, 19-1452, 19-1458, 2021 U.S. LEXIS 3124 (June 21, 2021), Patent Owner argues that “[u]ntil such time as procedures are in place allowing for [Director] review, the Board does not have authority to decide issues of patentability presented in a petition for *inter partes* review.” Prelim. Resp. 69.

Patent Owner’s argument is moot. The Office has implemented procedures for interim Director review following *Arthrex*. *See* <https://www.uspto.gov/patents/patent-trial-and-appeal-board/procedures/uspto-implementation-interim-director-review>. Specifically,

Parties may request Director review of a final written decision in an inter parties review or a post-grant review by concurrently (1) entering a Request for Rehearing by the Director into PTAB E2E and (2) submitting a notification of the Request for Rehearing by the Director to the Office by email to Director_PTABDecision_Review@uspto.gov, copying counsel for all parties by email.

IPR2021-00627

Patent 10,375,215 B1

Id.

CONCLUSION

Following 35 U.S.C. § 314, we have determined whether the totality of the information presented at this stage shows there is a reasonable likelihood that Petitioner would prevail with respect to at least one of the claims challenged in the Petition. And because Petitioner has demonstrated a reasonable likelihood of success in proving that at least one claim of the '215 patent is unpatentable, we institute on all grounds and all claims raised in the Petition.

Our factual findings, conclusions of law, and determinations at this stage of the proceeding are preliminary, and based on the evidentiary record developed thus far. This is not a final decision as to the patentability of claims for which *inter partes* review is instituted. Our final decision will be based on the record as fully developed during trial.

ORDER

In consideration of the foregoing, it is hereby:

ORDERED that, an *inter partes* review of all challenged claims of the '215 patent is instituted with respect to all grounds set forth in the Petition; and

FURTHER ORDERED that pursuant to 35 U.S.C. § 314(a), *inter partes* review of the '215 patent is hereby instituted commencing on the entry date of this Decision, and pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial.

IPR2021-00627

Patent 10,375,215 B1

FOR PETITIONER:

Naveen Modi

Joseph Palys

Quadeer Ahmed

Jason Heidemann

PAUL HASTINGS LLP

naveenmodi@paulhastings.com

josephpalys@paulhastings.com

quadeerahmed@paulhastings.com

jasonheidemann@paulhastings.com

FOR PATENT OWNER:

Timothy Devlin

DEVLIN LAW FIRM LLC

td-ptab@devlinlawfirm.com

EXHIBIT 18

JENAM TECH, LLC’S FIRST AMENDED SET OF INFRINGEMENT CONTENTIONS

U.S. Patent No. 10,306,026 – Google LLC Claims 22, 31, 32, 45, 46, 48, 51, 56, 57, 59, 60, 66, 67, and 69

Jenam Tech, LLC (“Jenam”) provides evidence of infringement of claims 22, 31, 32, 45, 46, 48, 51, 56, 57, 59, 60, 66, 67, and 69 of U.S. Patent No. 10,306,026 (hereinafter “the ’026 patent”) by Google LLC (“Google”). In support thereof, Jenam provides the following claim charts.

“Accused Instrumentalities” as used herein refers to at least one or more websites or web addresses including, but not limited to www.google.com, stored and/or hosted on one or more servers owned or under the control of Google. For the sake of clarity, Jenam alleges that all of Google’s various websites, products, and platforms utilizing QUIC infringe the ’026 patent, including, but not limited to Google Edge Network, Google Cloud, Chrome Enterprise, G suite, Google Play, Chrome, Android (Android Enterprise, Android Messages (RCS)), Duo, Google Ads, Adwords, Google Analytics, Youtube, Google Mobile apps, Google shopping, and Google Maps. A list of Google’s “products” can be found at https://about.google/intl/en_us/products/. On information and belief, the Accused Instrumentalities referred to in the charts below are representative of Google’s use of QUIC in Google’s other websites, products and platforms. These claim charts demonstrate Google’s infringement, and provide notice of such infringement, by comparing each element of the asserted claims to corresponding components, aspects, and/or features of the Accused Instrumentalities. These claim charts are not intended to constitute an expert report on infringement. These claim charts include information provided by way of example, and not by way of limitation.

The analysis set forth below is based only upon information from publicly available resources regarding the Infringing Instrumentalities. An analysis of Google’s (or other third parties’) technical documentation and/or software source code may assist in fully identify all infringing features and functionality. Accordingly, Jenam reserves the right to supplement this infringement analysis once such information is made available to Jenam. Furthermore, Jenam reserves the right to revise this infringement analysis, as appropriate, upon issuance of a court order construing any terms recited in the asserted claims. Unless otherwise noted, Jenam contends that Google directly infringes the ’026 patent in violation of 35 U.S.C. § 271(a) by selling, offering to sell, making, using, and/or importing the Infringing Instrumentalities. The following exemplary analysis demonstrates that infringement. Unless otherwise noted, Jenam further contends that the evidence below supports a finding of indirect infringement under 35 U.S.C. §§ 271(b) and/or (c), in conjunction with other evidence of liability under one or more of those subsections. Google makes, uses, sells, imports, or offers for sale in the United States, or has made, used, sold, imported, or offered for sale in the past, without authority, or induces others to make, use, sell, import, or offer for sale in the United States, or has induced others to make, use, sell, import, or offer for sale in the past, without authority products, equipment, or services that infringe claims 22, 31, 32, 45, 46, 48, 51, 56, 57, 59, 60, 66, 67, and 69 of the ’026 patent, including without limitation, the Accused Instrumentalities.

Unless otherwise noted, Jenam believes and contends that each element of each claim asserted herein is literally met through Google’s provision of the Infringing Instrumentalities. However, to the extent that Google attempts to allege that any asserted claim

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

element is not literally met, Jenam believes and contends that such elements are met under the doctrine of equivalents. More specifically, in its investigation and analysis of the Infringing Instrumentalities, Jenam did not identify any substantial differences between the elements of the patent claims and the corresponding features of the Infringing Instrumentalities, as set forth herein. In each instance, the identified feature of the Infringing Instrumentalities performs at least substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim element.

To the extent the chart of an asserted claim relies on evidence about certain specifically-identified Accused Instrumentalities, Jenam asserts that, on information and belief, any similarly-functioning instrumentalities also infringes the charted claim. Jenam reserves the right to amend this infringement analysis based on other products made, used, sold, imported, or offered for sale by Google. Jenam also reserves the right to amend this infringement analysis by citing other claims of the '026 patent, not listed in the claim chart, that are infringed by the Accused Instrumentalities. Jenam further reserves the right to amend this infringement analysis by adding, subtracting, or otherwise modifying content in the "Accused Instrumentalities" column of each chart.

Claim 1 Elements	Applicability
A method, comprising: at a node: receiving, from another node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including, at a node (e.g., one of a QUIC-compliant server or client, etc.): receiving, from another node (e.g., the other one of the QUIC-compliant server or client, etc.), a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their "Google Cloud Platform" uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>we're happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:

“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube”

https://en.wikipedia.org/wiki/Google_Cloud_Platform

Note: At least a portion of the citations herein are made to the QUIC standard found at:

<https://tools.ietf.org/html/draft-ietf-quic-transport-22>, and/or <https://tools.ietf.org/html/draft-ietf-quic-transport-27>. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:

<https://tools.ietf.org/html/draft-ietf-quic-transport-00>

<https://tools.ietf.org/html/draft-ietf-quic-transport-01>

<https://tools.ietf.org/html/draft-ietf-quic-transport-02>

<https://tools.ietf.org/html/draft-ietf-quic-transport-03>

<https://tools.ietf.org/html/draft-ietf-quic-transport-04>

<https://tools.ietf.org/html/draft-ietf-quic-transport-05>

<https://tools.ietf.org/html/draft-ietf-quic-transport-06>

<https://tools.ietf.org/html/draft-ietf-quic-transport-07>

<https://tools.ietf.org/html/draft-ietf-quic-transport-08>

<https://tools.ietf.org/html/draft-ietf-quic-transport-09>

<https://tools.ietf.org/html/draft-ietf-quic-transport-10>

<https://tools.ietf.org/html/draft-ietf-quic-transport-11>

<https://tools.ietf.org/html/draft-ietf-quic-transport-12>

<https://tools.ietf.org/html/draft-ietf-quic-transport-13>

<https://tools.ietf.org/html/draft-ietf-quic-transport-14>

<https://tools.ietf.org/html/draft-ietf-quic-transport-15>

<https://tools.ietf.org/html/draft-ietf-quic-transport-16>

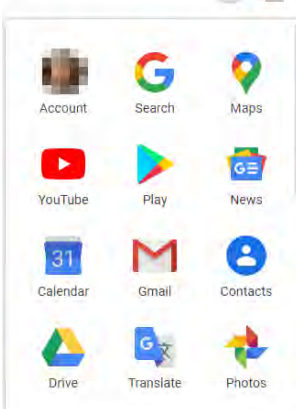
<https://tools.ietf.org/html/draft-ietf-quic-transport-17>

<https://tools.ietf.org/html/draft-ietf-quic-transport-18>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> <u>Note:</u> Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities. </p> <p> <u>Note:</u> Below is a web page of Google (https://www.google.com/). </p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

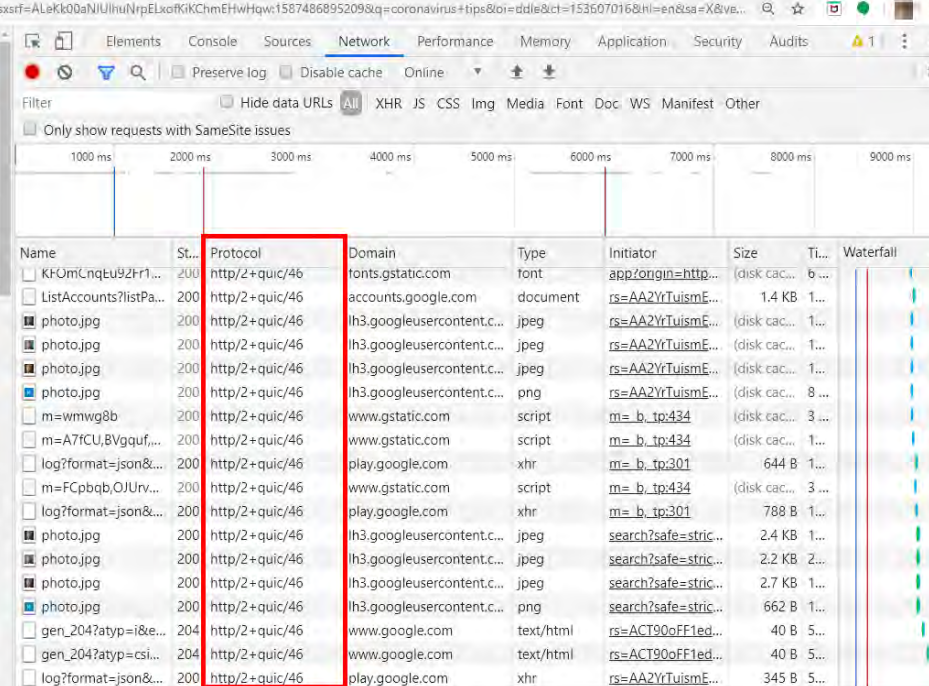


Cases overview

Alameda County

Confirmed	Recovered	Deaths
1,191	-	42

California



Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfUmCnqtu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app:origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_ to:434	(disk cac...	3...	
m=A7fCU,8Vgqf...	200	http/2+quic/46	www.gstatic.com	script	m= b_ to:434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ to:301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m= b_ to:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ to:301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, QUIC is a “variant” of TCP.

1. Introduction

QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.

QUIC implements techniques learned from experience with TCP, SCTP and

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” <https://www.chromium.org/quic>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>Note: As set forth below, a QUIC negotiation packet is received by one of the nodes.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>3.1. Low-Latency Connection Establishment</p> <p>QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].</p>
<p>detecting an idle time period parameter field in the TCP-variant packet;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs the method including, at the node: detecting an idle time period parameter field (e.g., idle timeout parameter field, etc.) in the TCP-variant packet (e.g., QUIC negotiation packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.

7.4. Transport Parameters

During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>The format of the transport parameters is the TransportParameters struct from Figure 6. This is described using the presentation language from Section 3 of [I-D.ietf-tls-tls13].</p> <pre> uint32 QuicVersion; enum { initial_max_stream_data(0), initial_max_data(1), initial_max_stream_id(2), idle_timeout(3), omit_connection_id(4), max_packet_size(5), stateless_reset_token(6), (65535) } TransportParameterId; </pre>
<p>identifying metadata in the idle time period parameter field for an idle time period and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active; and</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs the method including, at the node: identifying metadata (e.g., a value, etc.) in the idle time period parameter field (e.g., idle timeout parameter field, etc.) for an idle time period and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <pre> idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes). </pre>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p>
modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs the method including, at the node: modifying, based on the metadata (e.g., the value of the idle timeout parameter field, etc.), a timeout attribute associated with the TCP-variant connection.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</p> <p>7.8. Connection Termination</p> <p>Connections should remain open until they become idle for a pre-negotiated period of time. A QUIC connection, once established, can be terminated in one of three ways:</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 17 Elements	Applicability
<p>The method of claim 1 wherein one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.</p>	<p>Google infringes claim 1 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs the method wherein one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, infringement occurs, for example, at least during a packet communication scenario when a keep-alive packet starts a detected idle time period and no other keep-alives are communicated during the idle period.</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p style="text-align: center;">“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and</u> • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 21 Elements	Applicability
<p>The method of claim 17 wherein the keep-alive period is administered using a keep-alive timer.</p>	<p>Google infringes claim 17 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs the method wherein the keep-alive period is administered using a keep-alive timer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p>“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and</u> • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the <code>max_idle_timeouts</code> (see Section 18.2) and three times the current Probe Timeout (PTO).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p style="text-align: center;">“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 22 Elements	Applicability
<p>The method of claim 21 wherein the keep-alive timer is separate from an idle timer that is used to administer the idle time period.</p>	<p>Google infringes claim 21 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs the method wherein the keep-alive timer is separate from an idle timer that is used to administer the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires</u> (Section 10.2); and • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p style="text-align: center;">“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 31 Elements	Applicability
<p>The method of claim 1 wherein a duration of the idle time period is configured by an application via an analog of a socket.</p>	<p>Google infringes claim 1 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs the method wherein a duration of the idle time period is configured by an application via an analog of a socket (e.g., a listener socket and/or stream socket, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</u></p> <p>“The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

“Because of the built-in stream multiplexing layer in QUIC, applications need to handle three types of logical objects: listeners, connections, and streams. These objects have different properties from the traditional UDP and TCP listeners and connections. As a result, the adaptation to socket semantics was not direct. We used three types of sockets to expose the QUIC API, with the following properties:

- **Listener sockets:** like regular TCP/UDP listeners, they bind to a port and are used to accept incoming connections. Listeners can be closed when the application no longer desires to accept new connections.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<ul style="list-style-type: none">• Connection sockets: these sockets are created by connect or accept calls. Unlike TCP or UDP sockets, however, these sockets cannot be used to send or receive data and are only used to manage the streams belonging to the connection.• Stream sockets: these are the sockets that provide send and receive functions so peers can <u>exchange data</u>. They can be <u>created either by accepting on a connection socket (for streams opened by the peer), or by calling connect on the connection socket to create a new, locally initiated stream</u>. Streams can be closed by either peer at any point during the connection.” https://blogs.cisco.com/cloud/building-fast-quic-sockets-in-vpp (emphasis added)
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 32 Elements	Applicability
<p>The method of claim 1 wherein the metadata is identified by an application via an analog of a socket.</p>	<p>Google infringes claim 1 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs the method wherein the metadata (e.g. the value, etc.) is identified by an application via an analog of a socket (e.g., a listener socket and/or stream socket, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

7.4. Transport Parameters

During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

“Because of the built-in stream multiplexing layer in QUIC, applications need to handle three types of logical objects: listeners, connections, and streams. These objects have different properties from the

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>traditional UDP and TCP listeners and connections. As a result, the adaptation to socket semantics was not direct. <u>We used three types of sockets to expose the QUIC API</u>, with the following properties:</p> <ul style="list-style-type: none"> • <u>Listener sockets</u>: like regular TCP/UDP listeners, they bind to a port and are used to accept incoming connections. Listeners can be closed when the application no longer desires to accept new connections. • <u>Connection sockets</u>: these sockets are created by connect or accept calls. Unlike TCP or UDP sockets, however, these sockets cannot be used to send or receive data and are only used to manage the streams belonging to the connection. • <u>Stream sockets</u>: these are the sockets that provide send and receive functions so peers can exchange data. They can be created either by accepting on a connection socket (for streams opened by the peer), or by calling connect on the connection socket to create a new, locally initiated stream. Streams can be closed by either peer at any point during the connection.” <p>https://blogs.cisco.com/cloud/building-fast-quic-sockets-in-vpp (emphasis added)</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 33 Elements	Applicability
<p>A method, comprising: at a node: receiving idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method comprising: at a node: receiving idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection (e.g., the QUIC connection, etc.) to keep the TCP-variant connection active.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>we’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-00</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27) </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

(<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27>), QUIC_VERSION_IETF_DRAFT_28 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28>), QUIC_VERSION_IETF_DRAFT_29 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29>), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (<https://tools.ietf.org/html/draft-ietf-quic-invariants-06>) and/or IETF_QUIC_TRANSPORT_17 (<https://tools.ietf.org/html/draft-ietf-quic-transport-17>), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.

Note: Below is a web page of Google (<https://www.google.com/>).

Name	Size	Protocol	Domain	Type	Initiator	Size	Time	Waterfall
KfOmCnqLu9Zf1...	200	http/2+quic/46	fonts.gstatic.com	font	app/ongun=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_tp434	(disk cac...	3...	
m=A7fCU,BVgquf,...	200	http/2+quic/46	www.gstatic.com	script	m= b_tp434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_tp301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m= b_tp434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_tp301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, a QUIC is a “variant” of TCP.

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>“1. Introduction</p> <p>QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].</p> <p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the <code>idle_timeout</code> transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>generating a TCP-variant packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information; and</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including generating a TCP-variant packet (e.g., QUIC negotiation packet, etc.) including an idle time period parameter field (e.g., <code>idle_timeout</code> parameter field, etc.) identifying metadata (e.g., a value, etc.) for the idle time period based on the idle information.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

“18. Transport Parameter Encoding

The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].

enum {

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<pre> original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;" </pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>"19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the <code>idle_timeout</code> transport parameter (see Section 10). However, state in middleboxes</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>sending, from the node to another node, the TCP-variant packet in advance of the TCP-variant connection being established to provide the metadata for the idle time period to the another node, for use by the another node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including sending, from the node (e.g., one of a QUIC-compliant server or client, etc.) to another node (e.g., the other one of the QUIC-compliant server or client, etc.), the TCP-variant packet (e.g., QUIC negotiation packet, etc.) in advance of the TCP-variant connection (e.g., QUIC connection, etc.) being established to provide the metadata (e.g., the value, etc.) for the idle time period to the another node, for use by the another node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>“5.2.2. Server Packet Handling ... If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected. ...</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets.</p> <p>...</p> <p>7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 43 Elements	Applicability
<p>The method of claim 33 wherein one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.</p>	<p>Google infringes claim 33 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, infringement occurs, for example, at least during a packet communication scenario when a keep-alive packet starts a detected idle time period and no other keep-alives are communicated during the idle period.</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and</u> • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 45 Elements	Applicability
<p>The method of claim 43 wherein the keep-alive period is not configurable, while the idle time period is configurable.</p>	<p>Google infringes claim 43 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein the keep-alive period is not configurable, while the idle time period is configurable.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based on the list of negotiable parameters (see Section 18.2), the keep-alive period is not configurable during connection setup. Based upon information and belief, the keep-alive period is constrained by the idle time period, which is negotiated at connection setup.</p> <p>The duration for an idle time period includes negotiation/configuration via a max_idle_timeout transport parameter. See, for example, section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2spec and section 18.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-18.2). It is also clear from the list of negotiable parameters in section 18.2 that the keep-alive period is not negotiable/configurable during connection setup. Further, the keep-alive period is constrained by the idle time period as described in section 19.2 and 10.1.2. See section 10.1.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1 and section 19.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-19.2.</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p style="padding-left: 40px;">“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and</u> • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max idle timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 46 Elements	Applicability
<p>The method of claim 43 wherein the keep-alive period is administered using a keep-alive timer that is separate from an idle timer that is used to administer the idle time period.</p>	<p>Google infringes claim 43 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein the keep-alive period is administered using a keep-alive timer that is separate from an idle timer that is used to administer the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires</u> (Section 10.2); and • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 47 Elements	Applicability
<p>The method of claim 43 wherein the one or more keep-alive packets are conditionally communicated based on at least one keep-alive attribute.</p>	<p>Google infringes claim 43 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein the one or more keep-alive packets are conditionally communicated based on at least one keep-alive attribute.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“3.3. Session resumption versus Keep-alive</p> <p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p> <ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods. • Resume the session after a long idle period, using 0-RTT resumption when appropriate. <p>The first strategy is the easiest, but it only applies to certain applications. Either the server or the client in a QUIC application can send PING frames as keep-alives, to prevent the connection and any on-path state from timing out.</p> <p><u>Recommendations for the use of keep-alives are application specific, mainly depending on the latency requirements and message frequency of the application. In this case, the application mapping must specify whether the client or server is responsible for keeping the application alive.</u> Note that sending PING frames more frequently than every 30 seconds over long idle periods may result in a too much unproductive traffic and power usage for some situations.</p> <p><u>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic.</u> In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>restarting the connection. This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.</p> <p>The tradeoffs between resumption and keepalive need to be evaluated on a per-application basis. However, in general applications should use keepalives only in circumstances where continued communication is highly likely; [QUIC-HTTP], for instance, recommends using PING frames for keepalive only when a request is outstanding.”</p> <p>https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 48 Elements	Applicability
<p>The method of claim 47 wherein the keep-alive attribute is separate from the timeout attribute.</p>	<p>Google infringes claim 47 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein the keep-alive attribute is separate from the timeout attribute.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, the keep-alive attribute is satisfied by enabling or disabling the sending of one or more PING frames. See, for example, section 19.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-19.2 and section 10.1.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2.</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p style="padding-left: 40px;">“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and</u> • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 51 Elements	Applicability
<p>The method of claim 33 wherein a duration of the idle time period is configured by an application via an analog of a socket.</p>	<p>Google infringes claim 33 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein a duration of the idle time period is configured by an application via an analog of a socket (e.g., a listener socket and/or stream socket, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

7.4. Transport Parameters

During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

“Because of the built-in stream multiplexing layer in QUIC, applications need to handle three types of logical objects: listeners, connections, and streams. These objects have different properties from the

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>traditional UDP and TCP listeners and connections. As a result, the adaptation to socket semantics was not direct. <u>We used three types of sockets to expose the QUIC API</u>, with the following properties:</p> <ul style="list-style-type: none">• <u>Listener sockets</u>: like regular TCP/UDP listeners, they bind to a port and are used to accept incoming connections. Listeners can be closed when the application no longer desires to accept new connections.• <u>Connection sockets</u>: these sockets are created by connect or accept calls. Unlike TCP or UDP sockets, however, these sockets cannot be used to send or receive data and are only used to manage the streams belonging to the connection.• <u>Stream sockets</u>: these are the sockets that provide send and receive functions so peers can exchange data. They can be created either by accepting on a connection socket (for streams opened by the peer), or by calling connect on the connection socket to create a new, locally initiated stream. Streams can be closed by either peer at any point during the connection.” <p>https://blogs.cisco.com/cloud/building-fast-quic-sockets-in-vpp (emphasis added)</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 52 Elements	Applicability
<p>A method comprising: at a node configured to execute a network application such that the network application operates in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer:</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method comprising: at a node configured to execute a network application such that the network application operates in accordance with a non-transmission control protocol (TCP) protocol (e.g., a QUIC protocol, etc.) that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>we’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-00</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

(<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27>), QUIC_VERSION_IETF_DRAFT_28 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28>), QUIC_VERSION_IETF_DRAFT_29 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29>), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (<https://tools.ietf.org/html/draft-ietf-quic-invariants-06>) and/or IETF_QUIC_TRANSPORT_17 (<https://tools.ietf.org/html/draft-ietf-quic-transport-17>), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.

Note: Below is a web page of Google (<https://www.google.com/>).

Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfOmCnqLu9Zf1...	200	http/2+quic/46	fonts.gstatic.com	font	app/ongun=http...	(disk cac...	6 ...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8 ...	
m=wmg8b	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	3 ...	
m=A7fCU,BVgquf,...	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_tp301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	3 ...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_tp301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>“QUIC has been globally deployed at Google on thousands of servers and is used to serve traffic to a range of clients including a widely-used web browser (Chrome) . . .” https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/46403.pdf (pg 1)</p> <p>Note: As set forth below, a QUIC is a “variant” of TCP.</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“1. Introduction</p> <p>QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].</p> <p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.” <i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://medium.com/@nirosh/understanding-quic-wire-protocol-d0ff97644de7</p>
receiving, from another node, a non-TCP packet during a setup of a non-TCP connection;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including receiving, from another node (e.g., another one of the QUIC-compliant server or client, etc.), a non-TCP packet during a setup of a non-TCP connection (e.g., QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

“18. Transport Parameter Encoding

The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
identifying metadata, that specifies at least one of a number of seconds or minutes, in an idle time period parameter field in the non-TCP packet, for an idle time	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including identifying metadata (e.g., a value, etc.), that specifies at least one of a number of seconds or minutes, in an idle time period parameter field (e.g., idle_timeout parameter field, etc.) in the non-TCP packet (e.g., the QUIC negotiation packet, etc.), for an</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

<p>period, where, as a result of a detection of the idle time period, the non-TCP connection is subject to deactivation; and</p>	<p>idle time period, where, as a result of a detection of the idle time period, the non-TCP connection (e.g., the QUIC connection, etc.) is subject to deactivation (e.g., terminate the connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer and, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
determining, based on the metadata, a timeout attribute associated with the non-TCP connection;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including determining, based on the metadata (e.g., the value, etc.), a timeout attribute associated with the non-TCP connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the <code>idle_timeout</code> transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>wherein, based on operating in accordance with the TCP protocol, a three-way TCP handshake is performed for establishing a TCP connection that is different than the non-TCP connection.</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein, based on operating in accordance with the TCP protocol, a three-way TCP handshake is performed for establishing a TCP connection that is different than the non-TCP connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p><u>“To start a TCP connection 3 way handshake is performed.</u> This adds a significant delay when creating a new connection. We need to negotiate TLS, to create a secure, encrypted, https connection, even more network packets have to be sent back and forth.</p> <p>On the other hand UDP is connectionless. That means <u>UDP doesn’t establish connections as TCP does, so UDP does not perform this 3-way handshake</u> and for this reason, it is referred to as an unreliable protocol. That doesn’t mean UDP can’t transfer data, it just doesn’t negotiate how the connection will work, UDP just transmits and hopes for the best. The benefit is less time spent on the network to validate</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	packets, the downside is that in order to be reliable, something has to be built on top of UDP to confirm packet delivery.” https://medium.com/@nirosh/understanding-quic-wire-protocol-d0ff97644de7
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 55 Elements	Applicability
<p>The method of claim 52 wherein one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.</p>	<p>Google infringes claim 52 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, infringement occurs, for example, at least during a packet communication scenario when a keep-alive packet starts a detected idle time period and no other keep-alives are communicated during the idle period.</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and</u> • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 56 Elements	Applicability
<p>The method of claim 55 wherein the keep-alive period is not configurable, while the idle time period is configurable.</p>	<p>Google infringes claim 55 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein the keep-alive period is not configurable, while the idle time period is configurable.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based on the list of negotiable parameters (see Section 18.2), the keep-alive period is not configurable during connection setup. Based upon information and belief, the keep-alive period is constrained by the idle time period, which is negotiated at connection setup.</p> <p>The duration for an idle time period includes negotiation/configuration via a max_idle_timeout transport parameter. See, for example, section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2spec and section 18.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-18.2). It is also clear from the list of negotiable parameters in section 18.2 that the keep-alive period is not negotiable/configurable during connection setup. Further, the keep-alive period is constrained by the idle time period as described in section 19.2 and 10.1.2. See section 10.1.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1 and section 19.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-19.2.</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p style="padding-left: 40px;">“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires</u> (Section 10.2); and • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p style="text-align: center;">“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max idle timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 57 Elements	Applicability
<p>The method of claim 55 wherein the one or more keep-alive packets are conditionally communicated based on at least one keep-alive attribute.</p>	<p>Google infringes claim 55 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein the one or more keep-alive packets are conditionally communicated based on at least one keep-alive attribute.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, the keep-alive attribute is satisfied by enabling or disabling the sending of one or more PING frames. See, for example, section 19.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-19.2 and section 10.1.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2.</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval,</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows. <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“3.3. Session resumption versus Keep-alive</p> <p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p> <ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods. • Resume the session after a long idle period, using 0-RTT resumption when appropriate. <p>The first strategy is the easiest, but it only applies to certain applications. Either the server or the client in a QUIC application can send PING frames as keep-alives, to prevent the connection and any on-path state from timing out.</p> <p><u>Recommendations for the use of keep-alives are application specific</u>, mainly depending on the latency requirements and message frequency of the application. <u>In this case, the application mapping must specify whether the client or server is responsible for keeping the application alive</u>. Note that sending PING frames more frequently than every 30 seconds over long idle periods may result in a too much unproductive traffic and power usage for some situations.</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p><u>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic.</u> In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with restarting the connection. This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.</p> <p>The tradeoffs between resumption and keepalive need to be evaluated on a per-application basis. However, in general applications should use keepalives only in circumstances where continued communication is highly likely; [QUIC-HTTP], for instance, recommends using PING frames for keepalive only when a request is outstanding.”</p> <p>https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 59 Elements	Applicability
<p>The method of claim 55 wherein the node determines, based on the metadata, the timeout attribute associated with the non-TCP connection, in connection with negotiating, with the another node, a duration of the idle time period.</p>	<p>Google infringes claim 55 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein the node determines, based on the metadata (e.g., the value, etc.), the timeout attribute associated with the non-TCP connection (e.g., the QUIC connection, etc.), in connection with negotiating, with the another node, a duration of the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet is received by one of the nodes.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). <u>If this parameter is absent or zero then the idle timeout is disabled.</u>”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>“10.2. Idle Timeout</p> <p>...</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 60 Elements	Applicability
<p>The method of claim 55 wherein a duration of the idle time period is configured by an application via an analog of a socket.</p>	<p>Google infringes claim 55 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein a duration of the idle time period is configured by an application via an analog of a socket (e.g., a listener socket and/or stream socket, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

7.4. Transport Parameters

During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

“Because of the built-in stream multiplexing layer in QUIC, applications need to handle three types of logical objects: listeners, connections, and streams. These objects have different properties from the

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>traditional UDP and TCP listeners and connections. As a result, the adaptation to socket semantics was not direct. <u>We used three types of sockets to expose the QUIC API</u>, with the following properties:</p> <ul style="list-style-type: none">• <u>Listener sockets</u>: like regular TCP/UDP listeners, they bind to a port and are used to accept <u>incoming connections</u>. Listeners can be closed when the application no longer desires to accept new connections.• <u>Connection sockets</u>: these sockets are created by connect or accept calls. Unlike TCP or UDP sockets, however, these sockets cannot be used to send or receive data and are only used to manage the streams belonging to the connection.• <u>Stream sockets</u>: these are the sockets that provide send and receive functions so peers can <u>exchange data</u>. They can be <u>created either by accepting on a connection socket (for streams opened by the peer), or by calling connect on the connection socket to create a new, locally initiated stream</u>. Streams can be closed by either peer at any point during the connection.” <p>https://blogs.cisco.com/cloud/building-fast-quic-sockets-in-vpp (emphasis added)</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 61 Elements	Applicability
<p>A method comprising: at a node configured to execute a network application such that the network application is operates in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer:</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method comprising: at a node (e.g., one of a QUIC-compliant server or client, etc.) configured to execute a network application such that the network application is operates in accordance with a non-transmission control protocol (TCP) protocol (e.g., a QUIC protocol, etc.) that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>we’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-00</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

(<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27>), QUIC_VERSION_IETF_DRAFT_28 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28>), QUIC_VERSION_IETF_DRAFT_29 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29>), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (<https://tools.ietf.org/html/draft-ietf-quic-invariants-06>) and/or IETF_QUIC_TRANSPORT_17 (<https://tools.ietf.org/html/draft-ietf-quic-transport-17>), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.

Note: Below is a web page of Google (<https://www.google.com/>).

Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfOmCnqLu9Zf1...	200	http/2+quic/46	tonts.gstatic.com	font	app/ongun=http...	(disk cac...	6 ...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8 ...	
m=wmg8b	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	3 ...	
m=A7fCU,BVgquf,...	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_tp301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	3 ...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_tp301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, a QUIC is a “variant” of TCP.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>“1. Introduction</p> <p>QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].</p> <p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://medium.com/@nirosh/understanding-quic-wire-protocol-d0ff97644de7</p>
receiving idle information for use in detecting an idle time period that results in a non-TCP connection being subject to deactivation;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including receiving idle information for use in detecting an idle time period that results in a non-TCP connection (e.g., QUIC connection, etc.) being subject to deactivation.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO)."</p> <p>"The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
generating, based on the idle information, a non-TCP packet including an idle time period parameter field identifying metadata that is specified in at least one of a number of seconds or minutes; and	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including generating, based on the idle information, a non-TCP packet (e.g., QUIC negotiation packet, etc.) including an idle time period parameter field (e.g., idle_timeout parameter field, etc.) identifying metadata (e.g., a value, etc.) that is specified in at least one of a number of seconds or minutes.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>"7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>sending, from the node to another node and for establishing the non-TCP connection, the non-TCP packet to provide the metadata to the another node, for use by the another node in determining a timeout attribute associated with the non-TCP connection;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including sending, from the node (e.g., one of a QUIC-compliant server or client, etc.) to another node (e.g., the other one of the QUIC-compliant server or client, etc.) and for establishing the non-TCP connection (e.g., the QUIC connection, etc.), the non-TCP packet (e.g., the QUIC negotiation packet, etc.) to provide the metadata (e.g., the value, etc.) to the another node, for use by the another node in determining a timeout attribute associated with the non-TCP connection.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>“5.2.2. Server Packet Handling ... If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected. ... If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>...</p> <p>7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily determined based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
wherein a three-way TCP handshake is performed for establishing a TCP connection that is separate from the non-TCP connection.	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein a three-way TCP handshake is performed for establishing a TCP connection that is separate from the non-TCP connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the method (and webserver designed to operate therewith) supports both TCP (which is the current standard) and QUIC.</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"The standard way to do secure web browsing involves communicating over TCP + TLS, which requires 2 to 3 round trips with a server to establish a secure connection before the browser can request the actual web page ... Google has added QUIC support to recent builds of Chrome and enabled it for some of its online services, making it possible to test the protocol's real-world performance at scale."</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>https://www.theregister.co.uk/2015/04/17/google_quic_test_results/</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <ol style="list-style-type: none"> 1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3].”</p> <p>“Request for Comments” (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 64 Elements	Applicability
<p>The method of claim 61 wherein: one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period;</p>	<p>Google infringes claim 61 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein: one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, infringement occurs, for example, at least during a packet communication scenario when a keep-alive packet starts a detected idle time period and no other keep-alives are communicated during the idle period.</p> <p>Note: Based on the list of negotiable parameters (see Section 18.2), the keep-alive period is not configurable during connection setup. Based upon information and belief, the keep-alive period is constrained by the idle time period, which is negotiated at connection setup.</p> <p>The duration for an idle time period includes negotiation/configuration via a max_idle_timeout transport parameter. See, for example, section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2spec and section 18.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-18.2). It is also clear from the list of negotiable parameters in section 18.2 that the keep-alive period is not negotiable/configurable during connection setup. Further, the keep-alive period is constrained by the idle time period as described in section 19.2 and 10.1.2. See section 10.1.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1 and section 19.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-19.2.</p> <p>Note: Based upon information and belief, the keep-alive attribute is satisfied by enabling or disabling the sending of one or more PING frames. See, for example, section 19.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-19.2 and section 10.1.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p>“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and</u> • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the <code>max_idle_timeouts</code> (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a <code>max_idle_timeout</code>, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a <code>max_idle_timeout</code>, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

<p>the keep-alive period is not configurable, while the idle time period is configurable; and</p>	<p>Google infringes claim 61 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein the keep-alive period is not configurable, while the idle time period is configurable.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p style="padding-left: 40px;">“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires</u> (Section 10.2); and • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the <code>max_idle_timeouts</code> (see Section 18.2) and three times the current Probe Timeout (PTO).</p>
---	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
the one or more keep-alive packets are conditionally communicated based on at least one keep-alive attribute.	<p>Google infringes claim 61 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein the one or more keep-alive packets are conditionally communicated based on at least one keep-alive attribute.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“3.3. Session resumption versus Keep-alive</p> <p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p> <ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods. • Resume the session after a long idle period, using 0-RTT resumption when appropriate. <p>The first strategy is the easiest, but it only applies to certain applications. Either the server or the client in a QUIC application can send PING frames as keep-alives, to prevent the connection and any on-path state from timing out.</p> <p><u>Recommendations for the use of keep-alives are application specific</u>, mainly depending on the latency requirements and message frequency of the application. <u>In this case, the application mapping must specify whether the client or server is responsible for keeping the application alive.</u> Note that sending PING frames more frequently than every 30 seconds over long idle periods may result in a too much unproductive traffic and power usage for some situations.</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p><u>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic.</u> In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with restarting the connection. This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.</p> <p>The tradeoffs between resumption and keepalive need to be evaluated on a per-application basis. However, in general applications should use keepalives only in circumstances where continued communication is highly likely; [QUIC-HTTP], for instance, recommends using PING frames for keepalive only when a request is outstanding.”</p> <p>https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 66 Elements	Applicability
<p>The method of claim 64 and further comprising: configuring the node to cause the another node to determine, based on the metadata, the timeout attribute associated with the non-TCP connection, in connection with negotiating, with the node, a duration of the idle time period.</p>	<p>Google infringes claim 64 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method further comprising: configuring the node to cause the another node to determine, based on the metadata (e.g., the value, etc.), the timeout attribute associated with the non-TCP connection (e.g., the QUIC connection, etc.), in connection with negotiating, with the node, a duration of the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based on the list of negotiable parameters (see Section 18.2), the keep-alive period is not configurable during connection setup. Based upon information and belief, the keep-alive period is constrained by the idle time period, which is negotiated at connection setup.</p> <p>The duration for an idle time period includes negotiation/configuration via a max_idle_timeout transport parameter. See, for example, section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2spec and section 18.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-18.2). It is also clear from the list of negotiable parameters in section 18.2 that the keep-alive period is not negotiable/configurable during connection setup. Further, the keep-alive period is constrained by the idle time period as described in section 19.2 and 10.1.2. See section 10.1.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1 and section 19.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-19.2.</p> <p>Note: Based upon information and belief, the keep-alive attribute is satisfied by enabling or disabling the sending of one or more PING frames. See, for example, section 19.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-19.2 and section 10.1.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2.</p> <p>Note: As set forth below, a QUIC negotiation packet is received by one of the nodes.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). <u>If this parameter is absent or zero then the idle timeout is disabled.</u>”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>...</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-27.txt</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 67 Elements	Applicability
<p>The method of claim 64 wherein a duration of the idle time period is configured by an application via an analog of a socket.</p>	<p>Google infringes claim 64 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein a duration of the idle time period is configured by an application via an analog of a socket (e.g., a listener socket and/or stream socket, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based on the list of negotiable parameters (see Section 18.2), the keep-alive period is not configurable during connection setup. Based upon information and belief, the keep-alive period is constrained by the idle time period, which is negotiated at connection setup.</p> <p>The duration for an idle time period includes negotiation/configuration via a max_idle_timeout transport parameter. See, for example, section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2spec and section 18.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-18.2). It is also clear from the list of negotiable parameters in section 18.2 that the keep-alive period is not negotiable/configurable during connection setup. Further, the keep-alive period is constrained by the idle time period as described in section 19.2 and 10.1.2. See section 10.1.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1 and section 19.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-19.2.</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</u></p> <p>“The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>[QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

“Because of the built-in stream multiplexing layer in QUIC, applications need to handle three types of logical objects: listeners, connections, and streams. These objects have different properties from the traditional UDP and TCP listeners and connections. As a result, the adaptation to socket semantics was not direct. We used three types of sockets to expose the QUIC API, with the following properties:

- **Listener sockets:** like regular TCP/UDP listeners, they bind to a port and are used to accept incoming connections. Listeners can be closed when the application no longer desires to accept new connections.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<ul style="list-style-type: none">• Connection sockets: these sockets are created by connect or accept calls. Unlike TCP or UDP sockets, however, these sockets cannot be used to send or receive data and are only used to manage the streams belonging to the connection.• Stream sockets: these are the sockets that provide send and receive functions so peers can <u>exchange data</u>. They can be <u>created either by accepting on a connection socket (for streams opened by the peer), or by calling connect on the connection socket to create a new, locally initiated stream</u>. Streams can be closed by either peer at any point during the connection.” https://blogs.cisco.com/cloud/building-fast-quic-sockets-in-vpp (emphasis added)
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 68 Elements	Applicability
<p>A method comprising: at a node configured with a network application that is configured to operate in accordance with a first protocol including a transmission control protocol (TCP) to establish a TCP connection: based on operating in accordance with a second protocol, that is separate from the TCP, to establish a second protocol connection:</p> <p>receiving, from another node, a packet;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method comprising: at a node (e.g., one of a QUIC-compliant server or client, etc.) configured with a network application that is configured to operate in accordance with a first protocol including a transmission control protocol (TCP) to establish a TCP connection: based on operating in accordance with a second protocol (e.g., QUIC, etc.), that is separate from the TCP, to establish a second protocol connection (e.g., QUIC connection, etc.): receiving, from another node (e.g., the other one of the QUIC-compliant server or client, etc.), a packet (e.g., QUIC negotiation packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>we’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC</p>

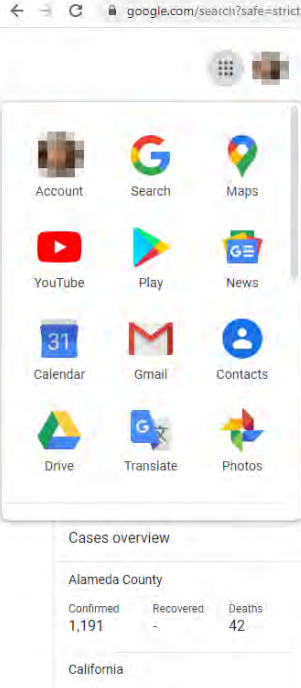
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

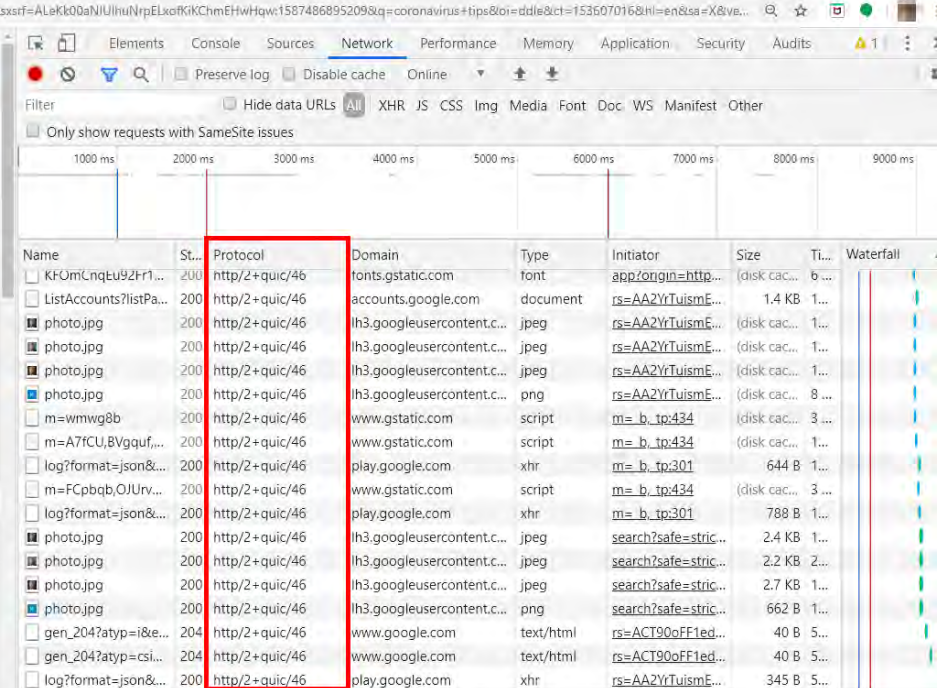
	<p>standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p> https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026





Note: As set forth below, QUIC is separate from TCP.

1. Introduction

QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.

QUIC implements techniques learned from experience with TCP, SCTP and

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>7.2. Version Negotiation</p> <p>QUIC’s connection establishment begins with version negotiation, since all communication between the endpoints, including packet and frame formats, relies on the two endpoints agreeing on a version.</p> <p>A QUIC connection begins with a client sending a Client Initial packet (Section 5.4.1). The details of the handshake mechanisms are described in Section 7.3, but all of the initial packets sent from the client to the server MUST use the long header format - which includes the version of the protocol being used - and they MUST be padded to at least 1200 octets.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. However, when the server generates a Version Negotiation packet, it cannot randomly generate a reserved version number. This is because the server is required to include the same value in its transport parameters (see Section 7.4.4). To avoid the selected version number</p> <ul style="list-style-type: none"> o The initial handshake packet from the client needs to fit in a single packet (#338) <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>3.1. Low-Latency Connection Establishment</p> <p>QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
detecting an idle time period parameter field in the packet;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including detecting an idle time period parameter field (e.g., idle timeout parameter field, etc.) in the packet (e.g., QUIC negotiation packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>The format of the transport parameters is the TransportParameters struct from Figure 6. This is described using the presentation language from Section 3 of [I-D.ietf-tls-tls13].</p> <pre>uint32 QuicVersion; enum { initial_max_stream_data(0), initial_max_data(1), initial_max_stream_id(2), idle_timeout(3), omit_connection_id(4), max_packet_size(5), stateless_reset_token(6), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>identifying metadata in the idle time period parameter field for an idle time period, where, in response to the idle time period being detected, the second protocol connection is deemed inactive; and</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including identifying metadata (e.g., a value, etc.) in the idle time period parameter field (e.g., idle timeout parameter field, etc.) for an idle time period, where, in response to the idle time period being detected, the second protocol connection (e.g., QUIC connection, etc.) is deemed inactive.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <pre>idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</pre>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
creating or modifying, based on the metadata, a timeout attribute associated with the second protocol connection.	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including creating or modifying, based on the metadata (e.g., the value, etc.), a timeout attribute associated with the second protocol connection (e.g., QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p><code>idle_timeout (0x0003):</code> The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,306,026

	<p>7.8. Connection Termination</p> <p>Connections should remain open until they become idle for a pre-negotiated period of time. A QUIC connection, once established, can be terminated in one of three ways:</p> <p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

Claim 69 Elements	Applicability
<p>The method of claim 68 wherein creating or modifying the timeout attribute renders one or more keep-alive packets in the second protocol connection unnecessary.</p>	<p>Google infringes claim 68 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method wherein creating or modifying the timeout attribute renders one or more keep-alive packets in the second protocol connection (e.g., QUIC connection, etc.) unnecessary.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO)."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>"19.2. PING Frame <u>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer.</u> The PING frame contains no additional fields.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"3.3. Session resumption versus Keep-alive</p> <p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p> <ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods. • Resume the session after a long idle period, using 0-RTT resumption when appropriate.
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,306,026

	<p>The first strategy is the easiest, but it only applies to certain applications. Either the server or the client in a QUIC application can send PING frames as keep-alives, to prevent the connection and any on-path state from timing out.</p> <p><u>Recommendations for the use of keep-alives are application specific, mainly depending on the latency requirements and message frequency of the application. In this case, the application mapping must specify whether the client or server is responsible for keeping the application alive.</u> Note that sending PING frames more frequently than every 30 seconds over long idle periods may result in a too much unproductive traffic and power usage for some situations.</p> <p><u>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic.</u> In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with restarting the connection. This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.</p> <p>The tradeoffs between resumption and keepalive need to be evaluated on a per-application basis. However, in general applications should use keepalives only in circumstances where continued communication is highly likely; [QUIC-HTTP], for instance, recommends using PING frames for keepalive only when a request is outstanding.”</p> <p>https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
--	---

Caveat: The notes and/or cited excerpts utilized herein are set forth for illustrative purposes only and are not meant to be limiting in any manner. For example, the notes and/or cited excerpts, may or may not be supplemented or substituted with different excerpt(s) of the relevant reference(s), as appropriate. Further, to the extent any error(s) and/or omission(s) exist herein, all rights are reserved to correct the same.

JENAM TECH, LLC’S FIRST AMENDED SET OF INFRINGEMENT CONTENTIONS

**U.S. Patent No. 10,375,215 – Google LLC
Claim 9**

Jenam Tech LLC (“Jenam”) provides evidence of infringement of claim 9 of U.S. Patent No. 10,375,215 (hereinafter “the ’215 patent”) by Google LLC (“Google”). In support thereof, Jenam provides the following claim charts.

“Accused Instrumentalities” as used herein refers to at least one or more websites or web addresses including, but not limited to www.google.com, stored and/or hosted on one or more servers owned or under the control of Google. For the sake of clarity, Jenam alleges that all of Google’s various websites, products, and platforms utilizing QUIC infringe the ’215 patent, including, but not limited to Google Edge Network, Google Cloud, Chrome Enterprise, G suite, Google Play, Duo, Google Ads, Adwords, Google Analytics, Youtube, Google Mobile apps, Google shopping, and Google Maps. A list of Google’s “products” can be found at https://about.google/intl/en_us/products/. On information and belief, the Accused Instrumentalities referred to in the charts below are representative of Google’s use of QUIC in Google’s other websites, products and platforms. These claim charts demonstrate Google’s infringement, and provide notice of such infringement, by comparing each element of the asserted claims to corresponding components, aspects, and/or features of the Accused Instrumentalities. These claim charts are not intended to constitute an expert report on infringement. These claim charts include information provided by way of example, and not by way of limitation.

The analysis set forth below is based only upon information from publicly available resources regarding the Infringing Instrumentalities. An analysis of Google’s (or other third parties’) technical documentation and/or software source code may assist in fully identify all infringing features and functionality. Accordingly, Jenam reserves the right to supplement this infringement analysis once such information is made available to Jenam. Furthermore, Jenam reserves the right to revise this infringement analysis, as appropriate, upon issuance of a court order construing any terms recited in the asserted claims.

Unless otherwise noted, Jenam contends that Google directly infringes the ’215 patent in violation of 35 U.S.C. § 271(a) by selling, offering to sell, making, using, and/or importing the Infringing Instrumentalities. The following exemplary analysis demonstrates that infringement. Unless otherwise noted, Jenam further contends that the evidence below supports a finding of indirect infringement under 35 U.S.C. §§ 271(b) and/or (c), in conjunction with other evidence of liability under one or more of those subsections. Google makes, uses, sells, imports, or offers for sale in the United States, or has made, used, sold, imported, or offered for sale in the past, without authority, or induces others to make, use, sell, import, or offer for sale in the United States, or has induced others to make, use, sell, import, or offer for sale in the past, without authority products, equipment, or services that infringe claim 9 of the ’215 patent, including without limitation, the Accused Instrumentalities.

Unless otherwise noted, Jenam believes and contends that each element of each claim asserted herein is literally met through Google’s provision of the Infringing Instrumentalities. However, to the extent that Google attempts to allege that any asserted claim element is not literally met, Jenam believes and contends that such elements are met under the doctrine of equivalents. More specifically, in its investigation and analysis of the Infringing Instrumentalities, Jenam did not identify any substantial differences

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

between the elements of the patent claims and the corresponding features of the Infringing Instrumentalities, as set forth herein. In each instance, the identified feature of the Infringing Instrumentalities performs at least substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim element.

To the extent the chart of an asserted claim relies on evidence about certain specifically-identified Accused Instrumentalities, Jenam asserts that, on information and belief, any similarly-functioning instrumentalities also infringes the charted claim. Jenam reserves the right to amend this infringement analysis based on other products made, used, sold, imported, or offered for sale by Google. Jenam also reserves the right to amend this infringement analysis by citing other claims of the '215 patent, not listed in the claim chart, that are infringed by the Accused Instrumentalities. Jenam further reserves the right to amend this infringement analysis by adding, subtracting, or otherwise modifying content in the "Accused Instrumentalities" column of each chart.

Claim 1 Elements	Applicability
<p>A computer-implemented method, comprising: at a server node: causing data to be sent to a client node; and providing access, to the client node, to code that, in response to being used by the client node, causes the client node to:</p> <p>receive, by the client node, a transmission control protocol (TCP)-variant packet,</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method, comprising: at a server node: causing data (e.g., in HTML pages, etc.) to be sent to a client node (e.g., device that receives the HTML pages, etc.); and providing access, to the client node, to code (e.g., also in the HTML pages, etc.) that, in response to being used by the client node, causes the client node to carry out the following functionality. Specifically, Google incorporates code into their HTML pages that, when used by the client node, causes the client node to communicate with one or more servers or at least one component thereof using the QUIC protocol.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their "Google Cloud Platform" uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>we're happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:

“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube”

https://en.wikipedia.org/wiki/Google_Cloud_Platform

Note: At least a portion of the citations herein are made to the QUIC standard found at:

<https://tools.ietf.org/html/draft-ietf-quic-transport-22>, and/or <https://tools.ietf.org/html/draft-ietf-quic-transport-27>. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:

<https://tools.ietf.org/html/draft-ietf-quic-transport-00>

<https://tools.ietf.org/html/draft-ietf-quic-transport-01>

<https://tools.ietf.org/html/draft-ietf-quic-transport-02>

<https://tools.ietf.org/html/draft-ietf-quic-transport-03>

<https://tools.ietf.org/html/draft-ietf-quic-transport-04>

<https://tools.ietf.org/html/draft-ietf-quic-transport-05>

<https://tools.ietf.org/html/draft-ietf-quic-transport-06>

<https://tools.ietf.org/html/draft-ietf-quic-transport-07>

<https://tools.ietf.org/html/draft-ietf-quic-transport-08>

<https://tools.ietf.org/html/draft-ietf-quic-transport-09>

<https://tools.ietf.org/html/draft-ietf-quic-transport-10>

<https://tools.ietf.org/html/draft-ietf-quic-transport-11>

<https://tools.ietf.org/html/draft-ietf-quic-transport-12>

<https://tools.ietf.org/html/draft-ietf-quic-transport-13>

<https://tools.ietf.org/html/draft-ietf-quic-transport-14>

<https://tools.ietf.org/html/draft-ietf-quic-transport-15>

<https://tools.ietf.org/html/draft-ietf-quic-transport-16>

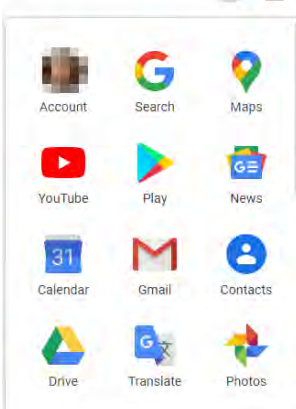
<https://tools.ietf.org/html/draft-ietf-quic-transport-17>

<https://tools.ietf.org/html/draft-ietf-quic-transport-18>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> <u>Note:</u> Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities. </p> <p> <u>Note:</u> Below is a web page of Google (https://www.google.com/) . </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

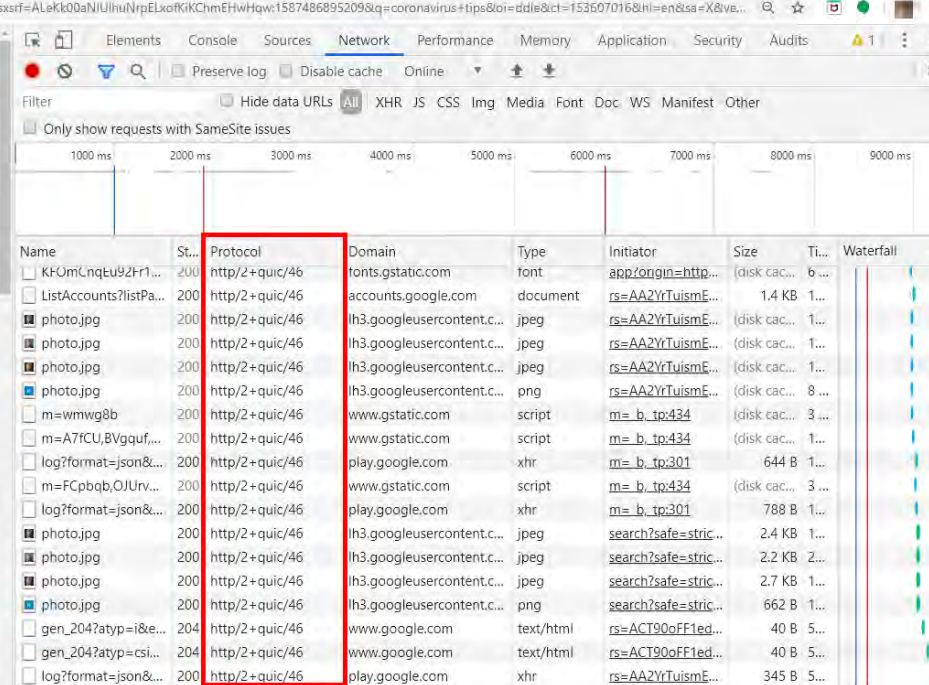


Cases overview

Alameda County

Confirmed	Recovered	Deaths
1,191	-	42

California



Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfUmCnqeu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app:origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
m=A7fCU,BVgqf...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	644 B	1...	
m=FCpbqb,OJUrv...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, QUIC is a “variant” of TCP.

1. Introduction

QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.

QUIC implements techniques learned from experience with TCP, SCTP and

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC negotiation packet is received by one of the nodes.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>3.1. Low-Latency Connection Establishment</p> <p>QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>detect an idle time period parameter field in the TCP-variant packet;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs the method including: at the server node: providing access, to the client node, to the code (e.g., in the HTML pages, etc.) that, in response to being used by the client node, causes the client node to: detect an idle time period parameter field (e.g., idle timeout parameter field, etc.) in the TCP-variant packet (e.g., QUIC negotiation packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>The format of the transport parameters is the TransportParameters struct from Figure 6. This is described using the presentation language from Section 3 of [I-D.ietf-tls-tls13].</p> <pre>uint32 QuicVersion; enum { initial_max_stream_data(0), initial_max_data(1), initial_max_stream_id(2), idle_timeout(3), omit_connection_id(4), max_packet_size(5), stateless_reset_token(6), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>identify metadata in the idle time period parameter field for an idle time period, during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active; and</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs the method including: at the server node: providing access, to the client node, to the code (e.g., in the HTML pages, etc.) that, in response to being used by the client node, causes the client node to: identify metadata (e.g., a value, etc.) in the idle time period parameter field (e.g., idle timeout parameter field, etc.) for an idle time period, during which, no packet is communicated in a TCP-variant connection to keep the TCP-variant connection active.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <pre>idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</pre>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
determine, based on the metadata, a timeout attribute associated with the TCP-variant connection.	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs the method including: at the server node: providing access, to the client node, to the code (e.g., in the HTML pages, etc.) that, in response to being used by the client node, causes the client node to: determine, based on the metadata (e.g., the value of the idle timeout parameter field, etc.), a timeout attribute associated with the TCP-variant connection.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>7.8. Connection Termination</p> <p>Connections should remain open until they become idle for a pre-negotiated period of time. A QUIC connection, once established, can be terminated in one of three ways:</p> <p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

Claim 4 Elements	Applicability
<p>The method of claim 1 wherein the code, to which the access is provided, is configured such that, in response to being used by the client node, causes the client node to operate such that one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.</p>	<p>Google infringes claim 1 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs the method including: at the server node: providing access, to the client node, to the code (e.g., in the HTML pages, etc.) that, in response to being used by the client node, causes the client node to: operate such that one or more keep-alive packets are communicated, based on a keep-alive period that is, in turn, based on the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p>“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and</u> • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max idle timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“3.3. Session resumption versus Keep-alive</p> <p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p> <ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods. • Resume the session after a long idle period, using 0-RTT resumption when appropriate.
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>The first strategy is the easiest, but it only applies to certain applications. Either the server or the client in a QUIC application can send PING frames as keep-alives, to prevent the connection and any on-path state from timing out.</p> <p><u>Recommendations for the use of keep-alives are application specific</u>, mainly depending on the latency requirements and message frequency of the application. <u>In this case, the application mapping must specify whether the client or server is responsible for keeping the application alive.</u> Note that sending PING frames more frequently than every 30 seconds over long idle periods may result in a too much unproductive traffic and power usage for some situations.</p> <p><u>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic.</u> In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with restarting the connection. This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.</p> <p>The tradeoffs between resumption and keepalive need to be evaluated on a per-application basis. However, in general applications should use keepalives only in circumstances where continued communication is highly likely; [QUIC-HTTP], for instance, recommends using PING frames for keepalive only when a request is outstanding.” https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p> <p>Note: Based upon information and belief, a keep-alive period for a connection that is longer than an idle time period for the connection would not prevent the connection from being deactivated as it wouldn't prevent detection of an idle time period and, thus, the keep-alive period is determined automatically by code based on the idle time period, and/or an application and/or a user is capable of providing input for the keep-alive period to be based on the idle time period.</p> <p>See IETF QUIC Transport section 10.1.2 on deferring an idle timeout utilizing a PING frame at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2. Also see, for example, section 10.1 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>10.1 and section 19.2 at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-19.2.</p> <p>"19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows"</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27</p> <p>"* keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and"</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,375,215

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

Claim 8 Elements	Applicability
<p>The method of claim 4 wherein the code, to which the access is provided, is configured such that, in response to being used by the client node, causes the client node to operate such that the keep-alive period is administered using a keep-alive timer.</p>	<p>Google infringes claim 4 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs the method including: at the server node: providing access, to the client node, to the code (e.g., in the HTML pages, etc.) that, in response to being used by the client node, causes the client node to: operate such that the keep-alive period is administered using a keep-alive timer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p>“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and</u> • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max idle timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

Claim 9 Elements	Applicability
<p>The method of claim 8 wherein the code, to which the access is provided, is configured such that, in response to being used by the client node, causes the client node to operate such that the keep-alive timer is separate from an idle timer that is used to administer the idle time period.</p>	<p>Google infringes claim 8 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs the method including: at the server node: providing access, to the client node, to the code (e.g., in the HTML pages, etc.) that, in response to being used by the client node, causes the client node to: operate such that the keep-alive timer is separate from an idle timer that is used to administer the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p>“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and</u> • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,375,215

	<p>guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max idle timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

Caveat: The notes and/or cited excerpts utilized herein are set forth for illustrative purposes only and are not meant to be limiting in any manner. For example, the notes and/or cited excerpts, may or may not be supplemented or substituted with different excerpt(s) of the relevant reference(s), as appropriate. Further, to the extent any error(s) and/or omission(s) exist herein, all rights are reserved to correct the same.

JENAM TECH LLC'S FIRST AMENDED SET OF INFRINGEMENT CONTENTIONS

**U.S. Patent No. 10,075,564 – Google LLC
Claims 23, 27, and 29**

Jenam Tech LLC (“Jenam”) provides evidence of infringement of claims 23, 27, and 29 of U.S. Patent No. 10,075,564 (hereinafter “the ’564 patent”) by Google LLC (“Google”). In support thereof, Jenam provides the following claim charts.

“Accused Instrumentalities” as used herein refers to at least one or more websites or web addresses including, but not limited to www.google.com, stored and/or hosted on one or more servers owned or under the control of Google. For the sake of clarity, Jenam alleges that all of Google’s various websites, products, and platforms utilizing QUIC infringe the ’564 patent, including, but not limited to Google Edge Network, Google Cloud, Chrome Enterprise, G suite, Google Play, Chrome, Android (Android Enterprise, Android Messages (RCS)), Duo, Google Ads, Adwords, Google Analytics, Youtube, Google Mobile apps, Google shopping, and Google Maps. A list of Google’s “products” can be found at https://about.google/intl/en_us/products/. On information and belief, the Accused Instrumentalities referred to in the charts below are representative of Google’s use of QUIC in Google’s other websites, products and platforms. These claim charts demonstrate Google’s infringement, and provide notice of such infringement, by comparing each element of the asserted claims to corresponding components, aspects, and/or features of the Accused Instrumentalities. These claim charts are not intended to constitute an expert report on infringement. These claim charts include information provided by way of example, and not by way of limitation.

The analysis set forth below is based only upon information from publicly available resources regarding the Infringing Instrumentalities. An analysis of Google’s (or other third parties’) technical documentation and/or software source code may assist in fully identify all infringing features and functionality. Accordingly, Jenam reserves the right to supplement this infringement analysis once such information is made available to Jenam. Furthermore, Jenam reserves the right to revise this infringement analysis, as appropriate, upon issuance of a court order construing any terms recited in the asserted claims.

Unless otherwise noted, Jenam contends that Google directly infringes the ’564 patent in violation of 35 U.S.C. § 271(a) by selling, offering to sell, making, using, and/or importing the Infringing Instrumentalities. The following exemplary analysis demonstrates that infringement. Unless otherwise noted, Jenam further contends that the evidence below supports a finding of indirect infringement under 35 U.S.C. §§ 271(b) and/or (c), in conjunction with other evidence of liability under one or more of those subsections. Google makes, uses, sells, imports, or offers for sale in the United States, or has made, used, sold, imported, or offered for sale in the past, without authority, or induces others to make, use, sell, import, or offer for sale in the United States, or has induced others to make, use, sell, import, or offer for sale in the past, without authority products, equipment, or services that infringe claims 23, 27, and 29 of the ’564 patent, including without limitation, the Accused Instrumentalities.

Unless otherwise noted, Jenam believes and contends that each element of each claim asserted herein is literally met through Google’s provision of the Infringing Instrumentalities. However, to the extent that Google attempts to allege that any asserted claim element is not literally met, Jenam believes and contends that such elements are met under the doctrine of equivalents. More

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

specifically, in its investigation and analysis of the Infringing Instrumentalities, Jenam did not identify any substantial differences between the elements of the patent claims and the corresponding features of the Infringing Instrumentalities, as set forth herein. In each instance, the identified feature of the Infringing Instrumentalities performs at least substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim element.

To the extent the chart of an asserted claim relies on evidence about certain specifically-identified Accused Instrumentalities, Jenam asserts that, on information and belief, any similarly-functioning instrumentalities also infringes the charted claim. Jenam reserves the right to amend this infringement analysis based on other products made, used, sold, imported, or offered for sale by Google. Jenam also reserves the right to amend this infringement analysis by citing other claims of the '564 patent, not listed in the claim chart, that are infringed by the Accused Instrumentalities. Jenam further reserves the right to amend this infringement analysis by adding, subtracting, or otherwise modifying content in the "Accused Instrumentalities" column of each chart.

Claim 16 Elements	Applicability
<p>A non-transitory computer readable medium, comprising:</p> <p>code for being communicated to a remote client node including one or more processors in communication with a non-transitory memory, where the code, when used by the client node, results in the client node operating to:</p> <p>identify idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable medium storing code (e.g., HTML pages, etc.) for being communicated to a remote client node (e.g., device that receives the HTML pages, etc.) including one or more processors in communication with a non-transitory memory, where the code, when used by the client node, causes the client node to carry out the following functionality. Specifically, Google incorporates code into their HTML pages that, when used by the client node, causes the client node to identify idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection (e.g., the QUIC connection, etc.) to keep the TCP-variant connection active.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their "Google Cloud Platform" uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>[W]e're happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p>

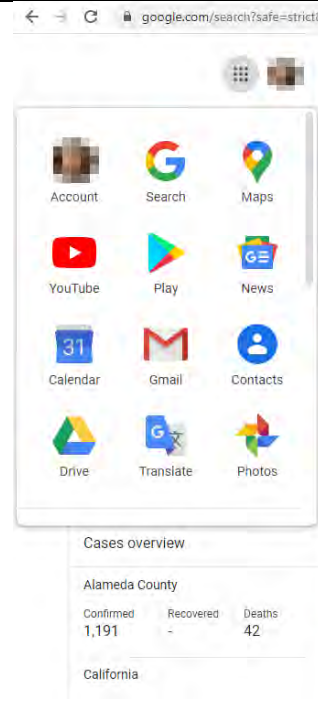
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

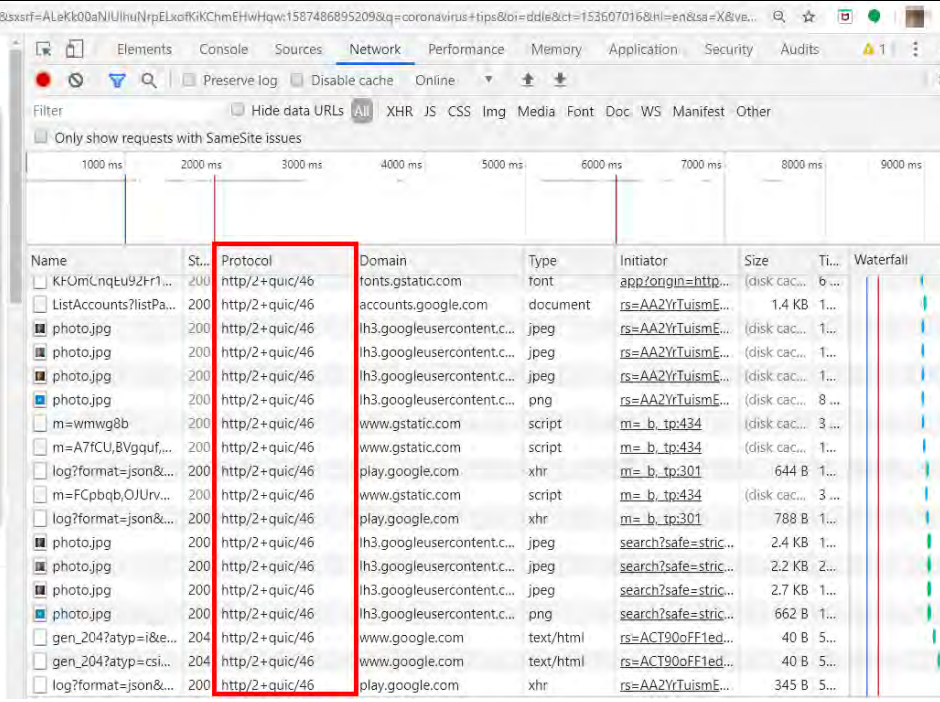
	<p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in a later version found at: https://tools.ietf.org/html/draft-ietf-quic-transport-27, as well as one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> <u>Note:</u> Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities. </p> <p> <u>Note:</u> Below is a web page of Google (https://www.google.com/). </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564





Name	Size	Protocol	Domain	Type	Initiator	Size	Time	Waterfall
KfUmCnqeu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app:origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
m=A7FCU,8Vgqf...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	644 B	1...	
m=FCpbqb,OJUrv...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, a QUIC is a “variant” of TCP.

“1. Introduction

QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].

QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.”

QUIC Loss Detection and Congestion Control <https://tools.ietf.org/html/draft-ietf-quic-recovery-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, there is “no activity” while the connection is idle.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the <code>idle_timeout</code> transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
generate a TCP-variant packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information; and	<p>Google incorporates code into their HTML pages that, when used by the client node, causes the client node to generate a TCP-variant packet (e.g., QUIC negotiation packet, etc.) including an idle time period parameter field (e.g., <code>idle_timeout</code> parameter field, etc.) identifying metadata (e.g., a value, etc.) for the idle time period based on the idle information; and</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the <code>idle_timeout</code> transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>send, from the client node to a server node, the TCP-variant packet to provide the metadata for the idle time period to the server node, for use by the server node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.</p>	<p>Google incorporates code into their HTML pages that, when used by the client node, causes the client node to send, from the client node to a server node, the TCP-variant packet (e.g., QUIC negotiation packet, etc.) to provide the metadata (e.g., the value, etc.) for the idle time period to the server node, for use by the server node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>“5.2.2. Server Packet Handling</p> <p>...</p> <p>If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected.</p> <p>...</p> <p>If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets.</p> <p>...</p> <p>7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,075,564

	<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the <code>idle_timeout</code> transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Claim 22 Elements	Applicability
<p>A system including the non-transitory computer readable medium of claim 16, wherein the system: includes another server node separate from the server node,</p> <p>is configured to communicate the code to the client node for use in establishing the TCP-variant connection with the server node, and</p> <p>is further configured to perform a 3-way TCP handshake for establishing, with the client node, a TCP connection that is different than the TCP-variant connection for permitting the client node to fetch the code in addition to other data from the system via a hypertext transfer protocol (HTTP).</p>	<p>Google infringes claim 16 and includes a system that incorporates code into their HTML pages wherein the system: includes another server (e.g., site, etc.) node separate from the server node, is configured to communicate the code to the client node for use in establishing the TCP-variant connection (e.g., the QUIC connection, etc.) with the server node, and is further configured to perform a 3-way TCP handshake for establishing, with the client node, a TCP connection that is different than the TCP-variant connection for permitting the client node to fetch the code in addition to other data from the system via a hypertext transfer protocol (HTTP).</p> <p>Note: In addition to the QUIC protocol, the system also operates in accordance with the standard transmission control protocol (TCP).</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“Google AdSense provides a way for publishers to earn money from their online content. AdSense works by matching ads to your site based on your content and visitors. ... You make your ad spaces available by pasting ad code on your site, and choose where you want the ads to appear.” https://support.google.com/adsense/answer/6242051?hl=en#zippy=%2Cwhat-is-adsense%2Cchow-does-adsense-differ-from-other-ad-networks</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side’s initial sequence number and send a confirming acknowledgment.</p> <ol style="list-style-type: none"> 1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3]."</p> <p>"Request for Comments" (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter that is detected by a recipient of such packet.</p> <p>"7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“3. Stream States</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p> <p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2].</u> This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state."</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Claim 23 Elements	Applicability
<p>The system of claim 22, wherein the code, when used by the client node, results in the client node operating such that the TCP-variant connection is established between the client node and the server node instead of the TCP connection in order to permit communication, between the client node and the server node, of the timeout attribute, where the timeout attribute is not communicated when establishing the TCP connection between the system and the client node, but is communicated when establishing the TCP-variant connection between the client and the server node.</p>	<p>Google infringes claim 22 and includes a system that incorporates code into their HTML pages wherein the code, when used by the client node, results in the client node operating such that the TCP-variant connection (e.g., the QUIC connection, etc.) is established between the client node and the server node instead of the TCP connection in order to permit communication, between the client node and the server node, of the timeout attribute, where the timeout attribute is not communicated when establishing the TCP connection between the system and the client node, but is communicated when establishing the TCP-variant connection between the client and the server node.</p> <p>Note: In addition to the QUIC protocol, the system also operates in accordance with the standard transmission control protocol (TCP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <ol style="list-style-type: none"> 1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,075,564

	<p>remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3].”</p> <p>“Request for Comments” (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Claim 24 Elements	Applicability
<p>A non-transitory computer readable medium, comprising:</p> <p>code for use by a client node including one or more processors in communication with a non-transitory memory storing a network application that is configured to operate in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, where the code, when used by the client node, results in the client node utilizing the network application to operate in accordance with the non-TCP protocol to:</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable medium storing code (e.g., HTML pages, etc.) for use by a client node (e.g., device that receives the HTML pages, etc.) including one or more processors in communication with a non-transitory memory, where the code, when used by the client node, causes the client node to carry out the following functionality. Specifically, Google incorporates code into their HTML pages that, when used by the client node, causes the client node to operate in accordance with a non-transmission control protocol (TCP) protocol (e.g., a QUIC protocol, etc.) that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, where the code, when used by the client node, results in the client node utilizing the network application to operate in accordance with the non-TCP protocol.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>[W]e’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22. Based upon information and belief, the</p>

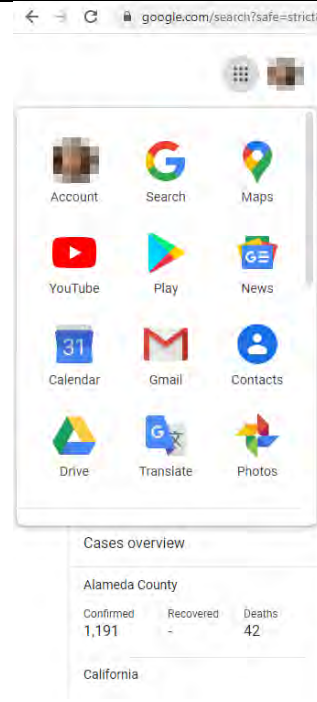
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>pertinent portions of such version of the QUIC standard may also be found in a later version found at: https://tools.ietf.org/html/draft-ietf-quic-transport-27, as well as one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p> https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,075,564

	<p><u>Note:</u> Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.</p> <p><u>Note:</u> Below is a web page of Google (https://www.google.com/).</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

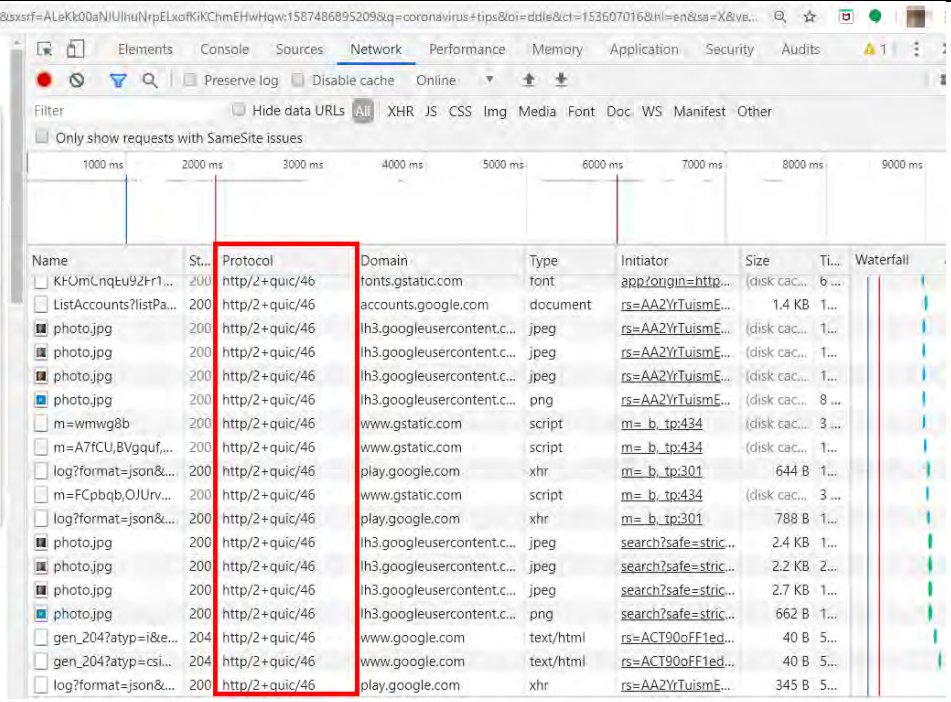


Cases overview

Alameda County

Confirmed	Recovered	Deaths
1,191	-	42

California



Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfUmCnqeu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app:origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
m=A7fCU,BVgqf...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, QUIC is a “variant” of TCP.

1. Introduction

QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.

QUIC implements techniques learned from experience with TCP, SCTP and

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC negotiation packet is received by one of the nodes.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>3.1. Low-Latency Connection Establishment</p> <p>QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>receive, from a server node, a non-TCP packet during a setup of a non-TCP connection;</p>	<p>Google incorporates code into their HTML pages that, when used by the client node, causes the client node to receive, from a server node, a non-TCP packet during a setup of a non-TCP connection (e.g., QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p><u>Note</u>: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

<p>identify metadata, that specifies a number of seconds or minutes, in an idle time period parameter field in the non-TCP packet, for an idle time period, where, as a result of a detection of the idle time period, the non-TCP connection is subject to deactivation; and</p>	<p>Google incorporates code into their HTML pages that, when used by the client node, causes the client node to identify metadata (e.g., a value, etc.), that specifies a number of seconds or minutes, in an idle time period parameter field (e.g., idle_timeout parameter field, etc.) in the non-TCP packet (e.g., the QUIC negotiation packet, etc.), for an idle time period, where, as a result of a detection of the idle time period, the non-TCP connection (e.g., the QUIC connection, etc.) is subject to deactivation (e.g., terminate the connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer and, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>
---	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO)."</p> <p>"The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>determine, based on the metadata, a timeout attribute associated with the non-TCP connection; wherein the code, when used by the client node, results in the client node utilizing the network application to operate in accordance with the TCP protocol to perform a three-way TCP handshake with another server node for setting up a TCP connection with the another server node that is different than the non-TCP connection.</p>	<p>Google incorporates code into their HTML pages that, when used by the client node, causes the client node to determine, based on the metadata (e.g., the value, etc.), a timeout attribute associated with the non-TCP connection (e.g., the QUIC connection, etc.); wherein the code, when used by the client node, results in the client node utilizing the network application to operate in accordance with the TCP protocol to perform a three-way TCP handshake with another server (e.g., site, etc.) node for setting up a TCP connection with the another server (e.g., site, etc.) node that is different than the non-TCP connection</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"Google AdSense provides a way for publishers to earn money from their online content. AdSense works by matching ads to your site based on your content and visitors. ... You make your ad spaces available by pasting ad code on your site, and choose where you want the ads to appear."</p> <p>https://support.google.com/adsense/answer/6242051?hl=en#zippy=%2Cwhat-is-adsense%2Cchow-does-adsense-differ-from-other-ad-networks</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p><u>Note:</u> The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p><u>“To start a TCP connection 3 way handshake is performed.</u> This adds a significant delay when creating a new connection. We need to negotiate TLS, to create a secure, encrypted, https connection, even more network packets have to be sent back and forth.</p> <p>On the other hand UDP is connectionless. That means <u>UDP doesn’t establish connections as TCP does, so UDP does not perform this 3-way handshake</u> and for this reason, it is referred to as an unreliable protocol. That doesn’t mean UDP can’t transfer data, it just doesn’t negotiate how the connection will work, UDP just transmits and hopes for the best. The benefit is less time spent on the network to validate packets, the downside is that in order to be reliable, something has to be built on top of UDP to confirm packet delivery.”</p> <p>https://medium.com/@nirosh/understanding-quic-wire-protocol-d0ff97644de7</p> <p><u>Note:</u> As set forth below, a QUIC negotiation packet includes transport parameters that include an <u>idle_timeout</u> parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,075,564

	<p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the <code>idle_timeout</code> transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Claim 25 Elements	Applicability
<p>A system including the another server node which includes the non-transitory computer readable medium of claim 24, wherein the system is configured to communicate the code to the client node utilizing the TCP connection, for setting up the non-TCP connection with the server node.</p>	<p>Google infringes claim 24 and includes a system that incorporates code into their HTML pages wherein the system is configured to communicate the code to the client node utilizing the TCP connection, for setting up the non-TCP connection (e.g., the QUIC connection, etc.) with the server node.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p><u>“To start a TCP connection 3 way handshake is performed.</u> This adds a significant delay when creating a new connection. We need to negotiate TLS, to create a secure, encrypted, https connection, even more network packets have to be sent back and forth.</p> <p>On the other hand UDP is connectionless. That means <u>UDP doesn’t establish connections as TCP does, so UDP does not perform this 3-way handshake</u> and for this reason, it is referred to as an unreliable protocol. That doesn’t mean UDP can’t transfer data, it just doesn’t negotiate how the connection will work, UDP just transmits and hopes for the best. The benefit is less time spent on the network to validate packets, the downside is that in order to be reliable, something has to be built on top of UDP to confirm packet delivery.”</p> <p>https://medium.com/@nirosh/understanding-quic-wire-protocol-d0ff97644de7</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Claim 26 Elements	Applicability
<p>The system of claim 25, wherein the system is configured to communicate the code to the client node via the TCP connection utilizing a hypertext transfer protocol (HTTP).</p>	<p>Google infringes claim 25 and includes a system that incorporates code into their HTML pages wherein the system is configured to communicate the code to the client node via the TCP connection utilizing a hypertext transfer protocol (HTTP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“3. Stream States</p> <p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p> <p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2].</u> This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state.”</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,075,564

--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Claim 27 Elements	Applicability
<p>The system of claim 26, wherein the code, when used by the client node, results in the client node operating such that the non-TCP connection is setup between the client node and the server node instead of the TCP connection in order to permit communication, between the client node and the server node, of the timeout attribute, where the timeout attribute is not communicated when setting up the TCP connection between the system and the client node, but is communicated when establishing the non-TCP connection between the client and the server node.</p>	<p>Google infringes claim 26 and includes a system that incorporates code into their HTML pages wherein the code, when used by the client node, results in the client node operating such that the non-TCP connection (e.g., the QUIC connection, etc.) is setup between the client node and the server node instead of the TCP connection in order to permit communication, between the client node and the server node, of the timeout attribute, where the timeout attribute is not communicated when setting up the TCP connection between the system and the client node, but is communicated when establishing the non-TCP connection between the client and the server node.</p> <p>Note: In addition to the QUIC protocol, the system also operates in accordance with the standard transmission control protocol (TCP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <p>1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y</p> <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,075,564

	<p>it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3].”</p> <p>“Request for Comments” (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Claim 28 Elements	Applicability
<p>A computer-implemented method, comprising: providing access to code for use by a client node including one or more processors in communication with a non-transitory memory storing a network application that is configured to operate in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, where the code causes the client node to utilize the network application to operate in accordance with the non-TCP protocol to:</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method comprising: providing access to code for use by a client node including one or more processors in communication with a non-transitory memory storing a network application that is configured to operate in accordance with a non-transmission control protocol (TCP) protocol (e.g., a QUIC protocol, etc.) that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, where the code causes the client node to utilize the network application to operate in accordance with the non-TCP protocol.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>[W]e’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in a later version found at: https://tools.ietf.org/html/draft-ietf-quic-transport-27, as well as one or more other versions. For</p>

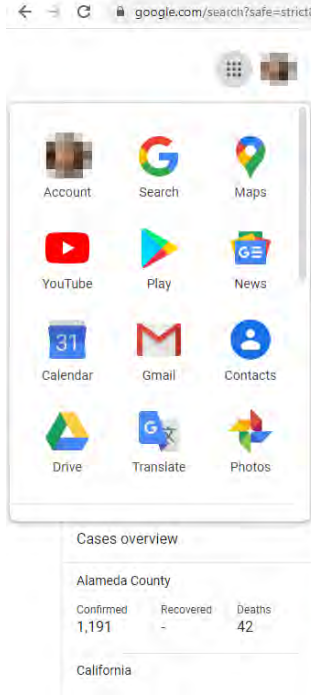
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

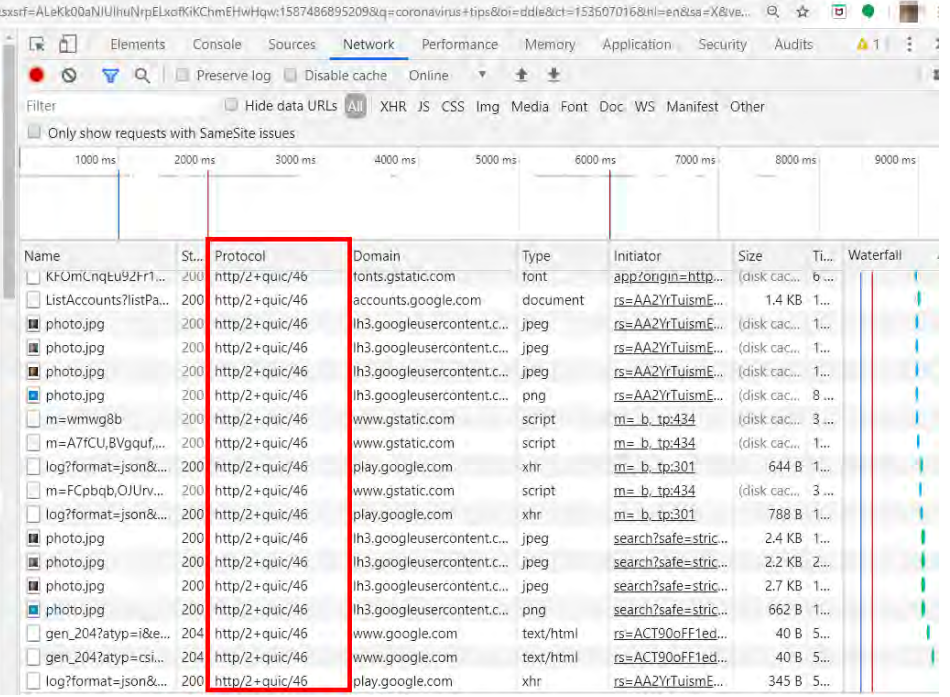
	<p>completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p> https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564





Name	Size	Protocol	Domain	Type	Initiator	Size	Time	Waterfall
Kf-UmLnqEu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app:origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
m=A7KCU,8Vgquf...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	644 B	1...	
m=FCpbqb,OJUrv...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, a QUIC is a “variant” of TCP.

“1. Introduction

QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].

QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.”

QUIC Loss Detection and Congestion Control <https://tools.ietf.org/html/draft-ietf-quic-recovery-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p><u>Note:</u> The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://medium.com/@nirosh/understanding-quic-wire-protocol-d0ff97644de7</p>
identify idle information for use in detecting an idle time period that results in a non-TCP connection being subject to deactivation;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method comprising: providing access to code for use by a client node including one or more processors in communication with a non-transitory memory storing a network application that is configured to identify idle information for use in detecting an idle time period that results in a non-TCP connection (e.g., QUIC connection, etc.) being subject to deactivation.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p><u>Note:</u> As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p><u>Note:</u> As set forth below, there is “no activity” while the connection is idle.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>generate, based on the idle information, a non-TCP packet including an idle time period parameter field identifying metadata that is specified in a number of seconds or minutes; and</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method comprising: providing access to code for use by a client node including one or more processors in communication with a non-transitory memory storing a network application that is configured to generate, based on the idle information, a non-TCP packet (e.g., QUIC negotiation packet, etc.) including an idle time period parameter field (e.g., idle_timeout parameter field, etc.) identifying metadata (e.g., a value, etc.) that is specified in a number of seconds or minutes.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p><u>Note:</u> As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

<p>send, from the client node to a server node and for setting up the non-TCP connection, the non-TCP packet to provide the metadata to the server node, for use by the server node in determining a timeout attribute associated with the non-TCP connection.</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method comprising: providing access to code for use by a client node including one or more processors in communication with a non-transitory memory storing a network application that is configured to send, from the client node to a server node and for setting up the non-TCP connection (e.g., the QUIC connection, etc.), the non-TCP packet (e.g., the QUIC negotiation packet, etc.) to provide the metadata (e.g., the value, etc.) to the server node, for use by the server node in determining a timeout attribute associated with the non-TCP connection.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>“5.2.2. Server Packet Handling ... If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected. ... If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets. ...</p> <p>7. Cryptographic and Transport Handshake</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,075,564

	<p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre> Enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;” </pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily determined based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the <code>idle_timeout</code> transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

Claim 29 Elements	Applicability
<p>The computer-implemented method of claim 28, and further comprising:</p> <p>providing access to additional code that causes the client node to utilize the network application to communicate with another server node in accordance with a TCP protocol using a hypertext transfer protocol (HTTP) and further using a TCP connection that is different than the non-TCP connection.</p>	<p>Google infringes claim 28 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method comprising: providing access to additional code that causes the client node to utilize the network application to communicate with another server (e.g., site, etc.) node in accordance with a TCP protocol using a hypertext transfer protocol (HTTP) and further using a TCP connection that is different than the non-TCP connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“Google AdSense provides a way for publishers to earn money from their online content. AdSense works by matching ads to your site based on your content and visitors. ... You make your ad spaces available by pasting ad code on your site, and choose where you want the ads to appear.” https://support.google.com/adsense/answer/6242051?hl=en#zippy=%2Cwhat-is-adsense%2Cchow-does-adsense-differ-from-other-ad-networks</p> <p>“3. Stream States</p> <p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p> <p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,564

	<p>model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2]</u>. This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state."</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)</p> <p>Note: In addition to the QUIC protocol, the method also operates in accordance with the standard transmission control protocol (TCP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <ol style="list-style-type: none"> 1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,075,564

	it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3]." "Request for Comments" (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793
--	--

Caveat: The notes and/or cited excerpts utilized herein are set forth for illustrative purposes only and are not meant to be limiting in any manner. For example, the notes and/or cited excerpts, may or may not be supplemented or substituted with different excerpt(s) of the relevant reference(s), as appropriate. Further, to the extent any error(s) and/or omission(s) exist herein, all rights are reserved to correct the same.

JENAM TECH LLC'S FIRST AMENDED SET OF INFRINGEMENT CONTENTIONS

**U.S. Patent No. 10,075,565 – Google LLC
Claims 25 and 28**

Jenam Tech LLC (“Jenam”) provides evidence of infringement of Claims 25 and 28 of U.S. Patent No. 10,075,565 (hereinafter “the ’565 patent”) by Google LLC (“Google”). In support thereof, Jenam provides the following claim charts.

“Accused Instrumentalities” as used herein refers to at least one or more websites or web addresses including, but not limited to www.google.com, stored and/or hosted on one or more servers owned or under the control of Google. For the sake of clarity, Jenam alleges that all of Google’s various websites, products, and platforms utilizing QUIC infringe the ’565 patent, including, but not limited to Google Edge Network, Google Cloud, Chrome Enterprise, G suite, Google Play, Duo, Google Ads, Adwords, Google Analytics, Youtube, Google Mobile apps, Google shopping, and Google Maps. A list of Google’s “products” can be found at https://about.google/intl/en_us/products/. On information and belief, the Accused Instrumentalities referred to in the charts below are representative of Google’s use of QUIC in Google’s other websites, products and platforms. These claim charts demonstrate Google’s infringement, and provide notice of such infringement, by comparing each element of the asserted claims to corresponding components, aspects, and/or features of the Accused Instrumentalities. These claim charts are not intended to constitute an expert report on infringement. These claim charts include information provided by way of example, and not by way of limitation.

The analysis set forth below is based only upon information from publicly available resources regarding the Infringing Instrumentalities. An analysis of Google’s (or other third parties’) technical documentation and/or software source code may assist in fully identify all infringing features and functionality. Accordingly, Jenam reserves the right to supplement this infringement analysis once such information is made available to Jenam. Furthermore, Jenam reserves the right to revise this infringement analysis, as appropriate, upon issuance of a court order construing any terms recited in the asserted claims.

Unless otherwise noted, Jenam contends that Google directly infringes the ’565 patent in violation of 35 U.S.C. § 271(a) by selling, offering to sell, making, using, and/or importing the Infringing Instrumentalities. The following exemplary analysis demonstrates that infringement. Unless otherwise noted, Jenam further contends that the evidence below supports a finding of indirect infringement under 35 U.S.C. §§ 271(b) and/or (c), in conjunction with other evidence of liability under one or more of those subsections. Google makes, uses, sells, imports, or offers for sale in the United States, or has made, used, sold, imported, or offered for sale in the past, without authority, or induces others to make, use, sell, import, or offer for sale in the United States, or has induced others to make, use, sell, import, or offer for sale in the past, without authority products, equipment, or services that infringe Claims 25 and 28 of the ’565 patent, including without limitation, the Accused Instrumentalities.

Unless otherwise noted, Jenam believes and contends that each element of each claim asserted herein is literally met through Google’s provision of the Infringing Instrumentalities. However, to the extent that Google attempts to allege that any asserted claim element is not literally met, Jenam believes and contends that such elements are met under the doctrine of equivalents. More specifically, in its investigation and analysis of the Infringing Instrumentalities, Jenam did not identify any substantial differences

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

between the elements of the patent claims and the corresponding features of the Infringing Instrumentalities, as set forth herein. In each instance, the identified feature of the Infringing Instrumentalities performs at least substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim element.

To the extent the chart of an asserted claim relies on evidence about certain specifically-identified Accused Instrumentalities, Jenam asserts that, on information and belief, any similarly-functioning instrumentalities also infringes the charted claim. Jenam reserves the right to amend this infringement analysis based on other products made, used, sold, imported, or offered for sale by Google. Jenam also reserves the right to amend this infringement analysis by citing other claims of the '565 patent, not listed in the claim chart, that are infringed by the Accused Instrumentalities. Jenam further reserves the right to amend this infringement analysis by adding, subtracting, or otherwise modifying content in the "Accused Instrumentalities" column of each chart.

Claim 25 Elements	Applicability
An apparatus comprising: a server computer including: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions such that a network application operates in accordance with a first protocol including a transmission control protocol (TCP), the server computer, when operating in accordance with the first protocol to set up a TCP connection, configured to:	<p>Google uses an apparatus (e.g., one or more server computers, and/or the Google Cloud Platform, etc.) including a non-transitory memory storing instructions (e.g., server and/or the Google Cloud Platform software, etc.), and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for carrying out the functionality set forth below.</p> <p>Note: In addition to the QUIC protocol, the apparatus also operates in accordance with the standard transmission control protocol (TCP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their "Google Cloud Platform" uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>we're happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The "Google Cloud Platform" that, as established above, uses QUIC, is also used for ALL of Google's services:</p>

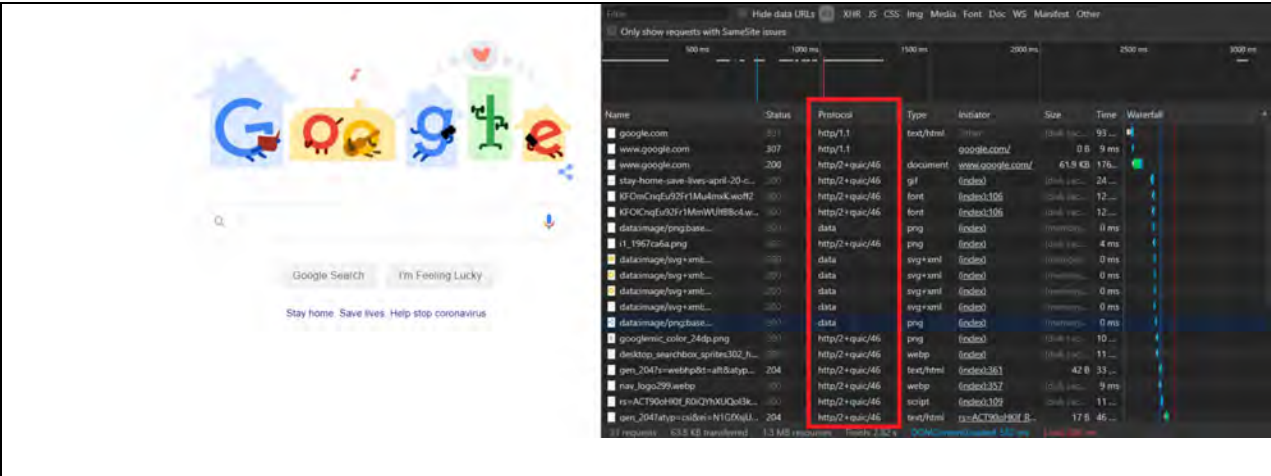
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

	<p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p> https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> <u>Note:</u> Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities. </p> <p> <u>Note:</u> Below is a web page of Google (https://www.google.com/). </p>
--	--

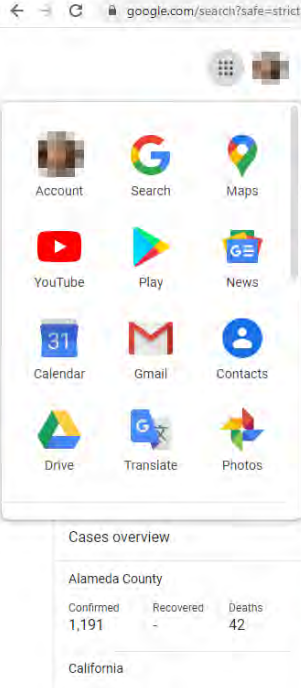
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

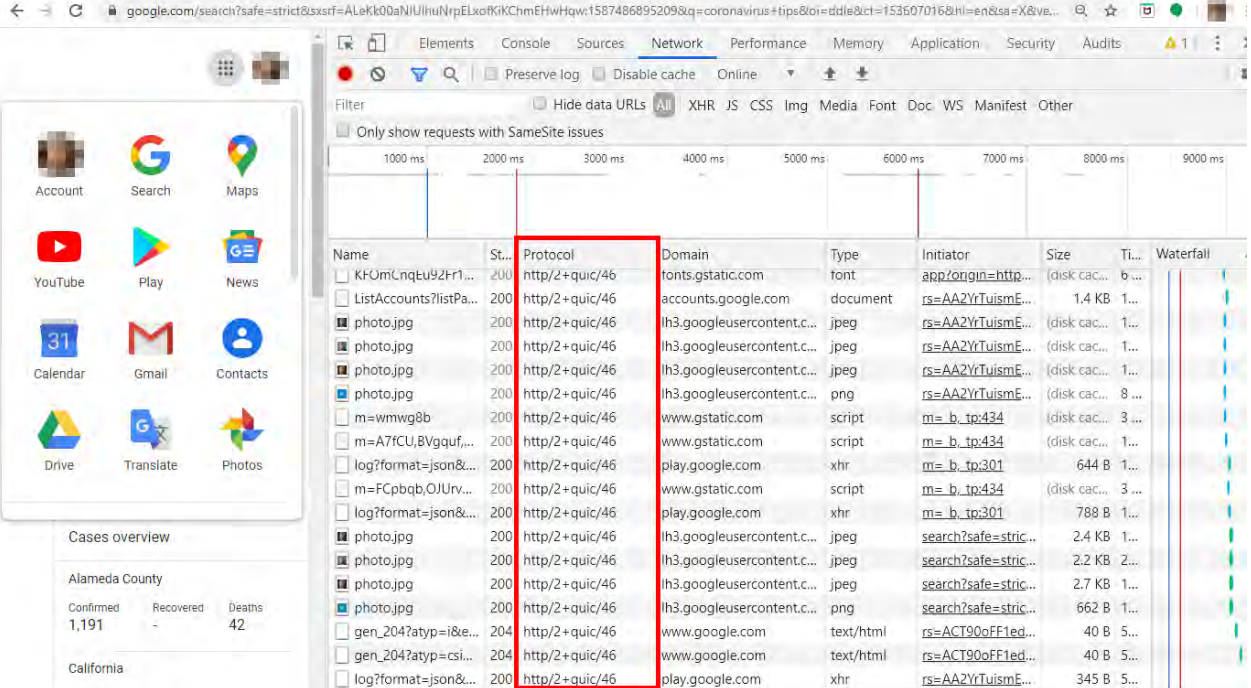
	
<p>communicate a segment including at least one first synchronize bit;</p> <p>communicate a first acknowledgement of the segment, and at least one second synchronize bit; and</p> <p>communicate a second acknowledgement;</p>	<p>Google uses the apparatus (e.g., the one or more server computers and/or the Google Cloud Platform, etc.) that, when operating in accordance with the first protocol to establish the TCP connection, is configured for a) communicating a segment including at least one first synchronize bit; b) communicating a first acknowledgement of the segment, and at least one second synchronize bit; and c) communicating a second acknowledgement.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a) is met by 1); b) is met by 2)/3), and c) is met by 4).</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <ol style="list-style-type: none"> 1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

	<p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3]."</p> <p>"Request for Comments" (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
<p>said server computer further configured to communicate, to a client computer, code that, when used by the client computer, causes the client computer to operate in accordance with a second protocol that is different from the TCP and that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, in order to setup a second protocol connection with another server computer, by:</p> <p>receiving, by the client computer from the another server computer, a packet;</p>	<p>Google uses the apparatus (e.g., the one or more server computers and/or the Google Cloud Platform, etc.) configured to communicate, to a client computer, code that, when used by the client computer, causes the client computer to operate in accordance with a second protocol (e.g., QUIC, etc.) that is different from the TCP and that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, in order to setup a second protocol connection (e.g., a QUIC connection, etc.) with another server (e.g., site, etc.) computer, by: receiving, by the client computer from the another server (e.g., site, etc.) computer, a packet (e.g., QUIC negotiation packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"Google AdSense provides a way for publishers to earn money from their online content. AdSense works by matching ads to your site based on your content and visitors. ... You make your ad spaces available by pasting ad code on your site, and choose where you want the ads to appear."</p> <p>https://support.google.com/adsense/answer/6242051?hl=en#zippy=%2Cwhat-is-adsense%2Cchow-does-adsense-differ-from-other-ad-networks</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565





Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfOmLnqEu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app/origin=http...	(disk cac...	6 ...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8 ...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_tp434	(disk cac...	3 ...	
m=A7fCU,BVgquf...	200	http/2+quic/46	www.gstatic.com	script	m= b_tp434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_tp301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m= b_tp434	(disk cac...	3 ...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_tp301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, QUIC is different from TCP.

1. Introduction

QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.

QUIC implements techniques learned from experience with TCP, SCTP and

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

	<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>7.2. Version Negotiation</p> <p>QUIC’s connection establishment begins with version negotiation, since all communication between the endpoints, including packet and frame formats, relies on the two endpoints agreeing on a version.</p> <p>A QUIC connection begins with a client sending a Client Initial packet (Section 5.4.1). The details of the handshake mechanisms are described in Section 7.3, but all of the initial packets sent from the client to the server MUST use the long header format - which includes the version of the protocol being used - and they MUST be padded to at least 1200 octets.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. However, when the server generates a Version Negotiation packet, it cannot randomly generate a reserved version number. This is because the server is required to include the same value in its transport parameters (see Section 7.4.4). To avoid the selected version number</p> <ul style="list-style-type: none"> o The initial handshake packet from the client needs to fit in a single packet (#338) <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

	<p>3.1. Low-Latency Connection Establishment</p> <p>QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>identifying metadata, that specifies a number of seconds or minutes, in an idle time period parameter field in the packet for an idle time period, during which, no packet is communicated that meets each of the following criteria: a) communicated via the second protocol connection, and b) causes the second protocol connection to be kept at least partially alive; and</p>	<p>Google uses the apparatus (e.g., the one or more server computers and/or the Google Cloud Platform, etc.) to identify metadata (e.g., a value, etc.), that specifies a number of seconds or minutes, in an idle time period parameter field (e.g., idle_timeout parameter field, etc.) in the packet for an idle time period and, during which, no packet is communicated that meets each of the following criteria: a) communicated via the second protocol connection (e.g., the QUIC connection, etc.), and b) causes the second protocol connection to be kept at least partially alive.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

	<p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

	<p>earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>determining, by the client computer and based on the metadata, a timeout attribute associated with the second protocol connection.</p>	<p>Google uses the apparatus (e.g., the one or more server computers and/or the Google Cloud Platform, etc.) to determine, by the client computer and based on the metadata (e.g., the value, etc.), a timeout attribute associated with the second protocol connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily determined based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

	<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO)."</p> <p>"The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>"19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows."</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,075,565

	<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

Claim 28 Elements	Applicability
<p>The apparatus of claim 25 wherein the code is communicated by permitting the client computer to fetch the code from the server computer utilizing the TCP connection and a hypertext transfer protocol (HTTP), and further wherein the code, when used by the client computer, causes the client computer to operate in accordance with the second protocol that is different from the TCP, by:</p> <p>detecting the idle time period based on the timeout attribute; and</p>	<p>Google infringes claim 25 and uses the apparatus (e.g., one or more server computer and/or the Google Cloud Platform s, etc.) wherein the code is communicated by permitting the client computer to fetch the code from the server computer utilizing the TCP connection and a hypertext transfer protocol (HTTP), and further wherein the code, when used by the client computer, causes the client computer to operate in accordance with the second protocol (e.g., QUIC, etc.) that is different from the TCP, by: detecting the idle time period based on the timeout attribute .</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“3. Stream States</p> <p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p> <p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2].</u> This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

	<p>mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state." https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)</p> <p>"idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled." <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>"10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO)." <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-27</p>
deactivating the second protocol connection by communicating a particular packet between the client computer and the another server computer, in response to detecting the idle time period.	<p>Google infringes claim 25 and uses the apparatus (e.g., one or more server computers and/or the Google Cloud Platform, etc.) to operate in accordance with the second protocol (e.g., QUIC, etc.) that is different from the TCP, by deactivating (e.g., terminating the connection, etc.) the second protocol connection (e.g., the QUIC connection, etc.) by communicating a particular packet between the client computer and the another server (e.g., site, etc.) computer, in response to detecting the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"Google AdSense provides a way for publishers to earn money from their online content. AdSense works by matching ads to your site based on your content and visitors. ... You make your ad spaces available by pasting ad code on your site, and choose where you want the ads to appear." https://support.google.com/adsense/answer/6242051?hl=en#zippy=%2Cwhat-is-adsense%2Cchow-does-adsense-differ-from-other-ad-networks</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,075,565

	<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,075,565

Caveat: The notes and/or cited excerpts utilized herein are set forth for illustrative purposes only and are not meant to be limiting in any manner. For example, the notes and/or cited excerpts, may or may not be supplemented or substituted with different excerpt(s) of the relevant reference(s), as appropriate. Further, to the extent any error(s) and/or omission(s) exist herein, all rights are reserved to correct the same.

JENAM TECH LLC'S FIRST AMENDED SET OF INFRINGEMENT CONTENTIONS

U.S. Patent No. 10,742,774 – Google LLC

Claims 3, 6, 7, 10, 11, 15, 16, 17, 19, 32, 36, 37, 38, 39, 44, 51, 59, 65, 67, and 72

Jenam Tech LLC (“Jenam”) provides evidence of infringement of claims 3, 6, 7, 10, 11, 15, 16, 17, 19, 32, 36, 37, 38, 39, 44, 51, 59, 65, 67, and 72 of U.S. Patent No. 10,742,774 (hereinafter “the ’774 patent”) by Google LLC (“Google”). In support thereof, Jenam provides the following claim charts.

“Accused Instrumentalities” as used herein refers to at least one or more websites or web addresses including, but not limited to www.google.com, stored and/or hosted on one or more servers owned or under the control of Google. For the sake of clarity, Jenam alleges that all of Google’s various websites, products, and platforms utilizing QUIC infringe the ’774 patent, including, but not limited to Google Edge Network, Google Cloud, Chrome Enterprise, G suite, Google Play, Chrome, Android (Android Enterprise, Android Messages (RCS)), Duo, Google Ads, Adwords, Google Analytics, Youtube, Google Mobile apps, Google shopping, and Google Maps. A list of Google’s “products” can be found at https://about.google/intl/en_us/products/. On information and belief, the Accused Instrumentalities referred to in the charts below are representative of Google’s use of QUIC in Google’s other websites, products and platforms. These claim charts demonstrate Google’s infringement, and provide notice of such infringement, by comparing each element of the asserted claims to corresponding components, aspects, and/or features of the Accused Instrumentalities. These claim charts are not intended to constitute an expert report on infringement. These claim charts include information provided by way of example, and not by way of limitation.

The analysis set forth below is based only upon information from publicly available resources regarding the Infringing Instrumentalities. An analysis of Google’s (or other third parties’) technical documentation and/or software source code may assist in fully identify all infringing features and functionality. Accordingly, Jenam reserves the right to supplement this infringement analysis once such information is made available to Jenam. Furthermore, Jenam reserves the right to revise this infringement analysis, as appropriate, upon issuance of a court order construing any terms recited in the asserted claims.

Unless otherwise noted, Jenam contends that Google directly infringes the ’774 patent in violation of 35 U.S.C. § 271(a) by selling, offering to sell, making, using, and/or importing the Infringing Instrumentalities. The following exemplary analysis demonstrates that infringement. Unless otherwise noted, Jenam further contends that the evidence below supports a finding of indirect infringement under 35 U.S.C. §§ 271(b) and/or (c), in conjunction with other evidence of liability under one or more of those subsections. Google makes, uses, sells, imports, or offers for sale in the United States, or has made, used, sold, imported, or offered for sale in the past, without authority, or induces others to make, use, sell, import, or offer for sale in the United States, or has induced others to make, use, sell, import, or offer for sale in the past, without authority products, equipment, or services that infringe claims 3, 6, 7, 10, 11, 15, 16, 17, 19, 32, 36, 37, 38, 39, 44, 51, 59, 65, 67, and 72 of the ’774 patent, including without limitation, the Accused Instrumentalities.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Unless otherwise noted, Jenam believes and contends that each element of each claim asserted herein is literally met through Google's provision of the Infringing Instrumentalities. However, to the extent that Google attempts to allege that any asserted claim element is not literally met, Jenam believes and contends that such elements are met under the doctrine of equivalents. More specifically, in its investigation and analysis of the Infringing Instrumentalities, Jenam did not identify any substantial differences between the elements of the patent claims and the corresponding features of the Infringing Instrumentalities, as set forth herein. In each instance, the identified feature of the Infringing Instrumentalities performs at least substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim element.

To the extent the chart of an asserted claim relies on evidence about certain specifically-identified Accused Instrumentalities, Jenam asserts that, on information and belief, any similarly-functioning instrumentalities also infringes the charted claim. Jenam reserves the right to amend this infringement analysis based on other products made, used, sold, imported, or offered for sale by Google. Jenam also reserves the right to amend this infringement analysis by citing other claims of the '774 patent, not listed in the claim chart, that are infringed by the Accused Instrumentalities. Jenam further reserves the right to amend this infringement analysis by adding, subtracting, or otherwise modifying content in the "Accused Instrumentalities" column of each chart.

Claim 1 Elements	Applicability
<p>An apparatus comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for:</p> <p>receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established;</p>	<p>Google owns, controls, and/or uses an apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for: receiving, by a second node (e.g., the other one of the QUIC-compliant server or client, etc.) from a first node (e.g., one of a QUIC-compliant server or client, etc.), a transmission control protocol (TCP)-variant packet (e.g., a QUIC negotiation packet, etc.) in advance of a TCP-variant connection (e.g., a QUIC connection, etc.) being established.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their "Google Cloud Platform" uses QUIC:</p> <p>"Google Cloud Platform Blog...</p>

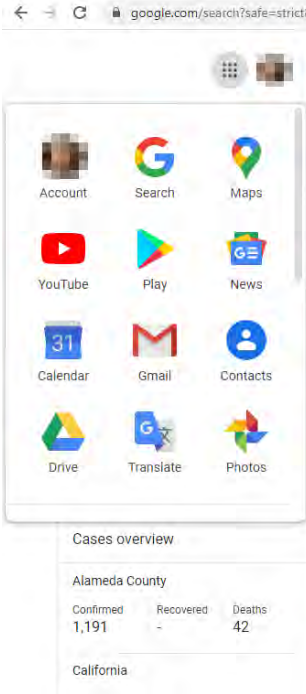
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>[W]e're happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The "Google Cloud Platform" that, as established above, uses QUIC, is also used for ALL of Google's services:</p> <p>"Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube" https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> <u>Note:</u> Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities. </p> <p> <u>Note:</u> Below is a web page of Google (https://www.google.com/). </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774



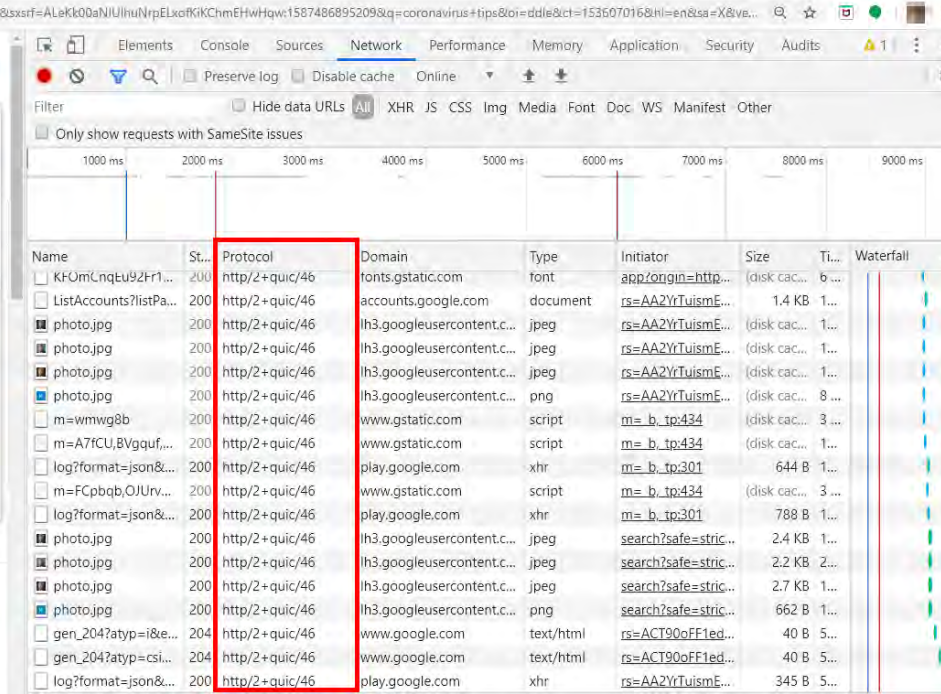
Account Search Maps
 YouTube Play News
 Calendar Gmail Contacts
 Drive Translate Photos

Cases overview

Alameda County

Confirmed	Recovered	Deaths
1,191	-	42

California



Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfOmCnqEu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app/origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_to:434	(disk cac...	3...	
m=A7fCU,BVgquf,...	200	http/2+quic/46	www.gstatic.com	script	m= b_to:434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_to:301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m= b_to:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_to:301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, QUIC is a “variant” of TCP.

1. Introduction

QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.

QUIC implements techniques learned from experience with TCP, SCTP and

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>7.2. Version Negotiation</p> <p>QUIC’s connection establishment begins with version negotiation, since all communication between the endpoints, including packet and frame formats, relies on the two endpoints agreeing on a version.</p> <p>A QUIC connection begins with a client sending a Client Initial packet (Section 5.4.1). The details of the handshake mechanisms are described in Section 7.3, but all of the initial packets sent from the client to the server MUST use the long header format - which includes the version of the protocol being used - and they MUST be padded to at least 1200 octets.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. However, when the server generates a Version Negotiation packet, it cannot randomly generate a reserved version number. This is because the server is required to include the same value in its transport parameters (see Section 7.4.4). To avoid the selected version number</p> <ul style="list-style-type: none"> o The initial handshake packet from the client needs to fit in a single packet (#338) <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>3.1. Low-Latency Connection Establishment</p> <p>QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
detecting a time period parameter field in the TCP-variant packet;	<p>Google owns, controls, and/or uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) with the one or more processors that execute the instructions for: detecting a time period parameter field (e.g., idle timeout parameter field, etc.) in the TCP-variant packet (e.g., QUIC negotiation packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>The format of the transport parameters is the TransportParameters struct from Figure 6. This is described using the presentation language from Section 3 of [I-D.ietf-tls-tls13].</p> <pre>uint32 QuicVersion; enum { initial_max_stream_data(0), initial_max_data(1), initial_max_stream_id(2), idle_timeout(3), omit_connection_id(4), max_packet_size(5), stateless_reset_token(6), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>identifying metadata in the time period parameter field for a time period, during which no packet is received from the first node in a data stream of the TCP-variant connection and processed by the second node to keep the TCP-variant connection active; and</p>	<p>Google owns, controls, and/or uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) with the one or more processors that execute the instructions for: identifying metadata (e.g., a value of the idle timeout parameter field, etc.) in the time period parameter field (e.g., idle timeout parameter field, etc.) for a time period, during which no packet is received from the first node (e.g., one of the QUIC-compliant server or client, etc.) in a data stream of the TCP-variant connection (e.g., the QUIC connection, etc.) and processed by the second node (e.g., the other one of the QUIC-compliant server or client, etc.) to keep the TCP-variant connection active.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</p> <p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>calculating, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection for being used to at least partially close the TCP-variant connection.</p>	<p>Google owns, controls, and/or uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) with the one or more processors that execute the instructions for: calculating, by the second node (e.g., the other one of the QUIC-compliant server or client, etc.) and based on the metadata (e.g., the value of the idle timeout parameter field, etc.), a timeout attribute associated with the TCP-variant connection (e.g., the QUIC connection, etc.) for being used to at least partially close the TCP-variant connection.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</p> <p>7.8. Connection Termination</p> <p>Connections should remain open until they become idle for a pre-negotiated period of time. A QUIC connection, once established, can be terminated in one of three ways:</p> <p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"5. Connection Closure</p> <p>Once established, an HTTP/3 connection can be used for many requests and responses over time until the connection is closed. <u>Connection closure can happen in any of several different ways.</u></p> <p>5.1. Idle Connections</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p><u>Each QUIC endpoint declares an idle timeout during the handshake. If the connection remains idle (no packets received) for longer than this duration, the peer will assume that the connection has been closed.</u> HTTP/3 implementations will need to open a new connection for new requests if the existing connection has been idle for longer than the server's advertised idle timeout, and SHOULD do so if approaching the idle timeout.</p> <p>HTTP clients are expected to request that the transport keep connections open while there are responses outstanding for requests or server pushes, as described in Section 10.2.2 of [QUIC-TRANSPORT]. <u>If the client is not expecting a response from the server, allowing an idle connection to time out is preferred over expending effort maintaining a connection that might not be needed.</u> A gateway MAY maintain connections in anticipation of need rather than incur the latency cost of connection establishment to servers. <u>Servers SHOULD NOT actively keep connections open.</u>" https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 3 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured such that, during the time period, no packet is received from the first node in the data stream of the TCP-variant connection regardless whether being processed by the second node.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that, during the time period, no packet is received from the first node (e.g., one of the QUIC-compliant server or client, etc.) in the data stream of the TCP-variant connection (e.g., the QUIC connection, etc.) regardless whether being processed by the second node (e.g., the other one of the QUIC-compliant server or client, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“19.2. PING Frame <u>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer.</u> The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle timeout transport parameter</u> (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 6 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured such that the timeout attribute associated with the TCP-variant connection is capable of being used by the second node to unilaterally cause the TCP-variant connection to at least partially close.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that the timeout attribute associated with the TCP-variant connection (e.g., the QUIC connection, etc.) is capable of being used by the second node (e.g., the other one of the QUIC-compliant server or client, etc.) to unilaterally cause the TCP-variant connection to at least partially close.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values. By announcing a max_idle_timeout, <u>an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</u>”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 7 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured such that, during the time period, no packet is received, nor expected to be received, from the first node in the data stream of the TCP-variant connection.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that, during the time period, no packet is received, nor expected to be received, from the first node (e.g., one of the QUIC-compliant server or client, etc.) in the data stream of the TCP-variant connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“19.2. PING Frame <u>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer.</u> The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle timeout transport parameter</u> (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 10 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured such that a user timeout attribute, that is separate from the timeout attribute, is utilized for controlling communication of one or more acknowledgment packets.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that a user timeout attribute, that is separate from the timeout attribute, is utilized for controlling communication of one or more acknowledgment packets (e.g., responding to user timeout event within a transmission control block (TCB) designed to hold connection state information, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, the user timeout attribute in QUIC is based on the max_ack_delay transport parameter. See section 4.7 in the IETF QUIC recovery draft at https://tools.ietf.org/html/draft-ietf-quic-recovery-34#section-4.7. The definition of max_ack_delay in the QUIC transport establishes that the user timeout is used to control ACK packets (copied below). See also section 13.2.1 "Sending ACK Frames" in the IETF QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-13.2.1.</p> <p>"max_ack_delay (0x0b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid." https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-18.2</p> <p>"13.2.1. Sending ACK Frames</p> <p>Every packet SHOULD be acknowledged at least once, and ack-eliciting packets MUST be acknowledged at least once within the maximum ack delay. An endpoint communicates its maximum delay using the max_ack_delay transport parameter; see Section 18.2. max_ack_delay declares an explicit contract: an endpoint promises to never intentionally delay acknowledgments of an ack-eliciting packet by more than the indicated value. If it does, any excess accrues to the RTT estimate and could result in spurious or delayed retransmissions from the peer. For Initial and Handshake packets, a max_ack_delay of 0 is used. The sender uses the receiver's "max_ack_delay"</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>value in determining timeouts for timer-based retransmission, as detailed in Section 5.2.1 of [QUIC-RECOVERY]."</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p><u>"3.9. Event Processing</u></p> <p>The processing depicted in this section is an example of one possible implementation. Other implementations may have slightly different processing sequences, but they should differ from those in this section only in detail, not in substance.</p> <p><u>The activity of the TCP can be characterized as responding to events.</u> The events that occur can be cast into three categories: user calls, arriving segments, and <u>timeouts</u>. This section describes the processing the TCP does in response to each of the events. In many cases the processing required depends on the state of the connection.</p> <p>...</p> <p>Timeouts</p> <p><u>USER TIMEOUT</u> RETRANSMISSION TIMEOUT TIME-WAIT TIMEOUT</p> <p>...</p> <p>OPEN Call</p> <p>CLOSED STATE (i.e., TCB does not exist)</p> <p><u>Create a new transmission control block (TCB) to hold connection state information. Fill in local socket identifier, foreign socket, precedence, security/compartments, and user timeout information.</u> Note that some parts of the foreign socket may be unspecified in a passive OPEN and are to be filled in by the parameters of the incoming SYN segment. Verify the security and precedence requested are allowed for this user, if not return "error: precedence not allowed" or "error: security/compartments not allowed." If passive enter the LISTEN state and return. If active and the</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,742,774

	foreign socket is unspecified, return "error: foreign socket unspecified"; if active and the foreign socket is specified, issue a SYN segment. An initial send sequence number (ISS) is selected. A SYN segment of the form <SEQ=ISS><CTL=SYN> is sent. Set SND.UNA to ISS, SND.NXT to ISS+1, enter SYN-SENT state, and return." https://tools.ietf.org/html/rfc793 (emphasis added)
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 11 Elements	Applicability
<p>The apparatus of claim 10 wherein the apparatus is configured such that the user timeout attribute is determined based on the time period.</p>	<p>Google infringes claim 10 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that the user timeout attribute is determined based on the time period.</p> <p>Note: Based upon information and belief, the user timeout in QUIC is based on the max_ack_delay transport parameter negotiated during setup. It is utilized to determine a probe timeout (PTO). Note that the max_ack_delay parameter is negotiated during setup. See QUIC transport draft section 13.2.1 at Hypertext Transfer Protocol Version 3 (HTTP/3) (ietf.org), section 4.7 in the IETF QUIC recovery draft at https://tools.ietf.org/html/draft-ietf-quic-recovery-34#section-4.7, and section 10.1 in the IETF QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“Case 2: TCP receives a FIN from the network</p> <p>If an unsolicited FIN arrives from the network, the receiving TCP can ACK it and tell the user that the connection is closing. The user will respond with a CLOSE, upon which the TCP can send a FIN to the other TCP after sending any remaining data. <u>The TCP then waits until its own FIN is acknowledged whereupon it deletes the connection. If an ACK is not forthcoming, after the user timeout the connection is aborted and the user is told.</u></p> <p>...</p> <p>3.9. <u>Event Processing</u></p> <p>The processing depicted in this section is an example of one possible implementation. Other implementations may have slightly different processing sequences, but they should differ from those in this section only in detail, not in substance.</p> <p><u>The activity of the TCP can be characterized as responding to events.</u> The events that occur can be cast into three categories: user calls, arriving segments, and <u>timeouts</u>. This section describes the processing the TCP does in response to each of the events. In many cases the processing required depends on the state of the connection.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>...</p> <p>Timeouts</p> <p><u>USER TIMEOUT</u></p> <p>RETRANSMISSION TIMEOUT</p> <p>TIME-WAIT TIMEOUT</p> <p>...</p> <p>OPEN Call</p> <p>CLOSED STATE (i.e., TCB does not exist)</p> <p><u>Create a new transmission control block (TCB) to hold connection state information. Fill in local socket identifier, foreign socket, precedence, security/compartments, and user timeout information.</u> Note that some parts of the foreign socket may be unspecified in a passive OPEN and are to be filled in by the parameters of the incoming SYN segment. Verify the security and precedence requested are allowed for this user, if not return "error: precedence not allowed" or "error: security/compartments not allowed." If passive enter the LISTEN state and return. If active and the foreign socket is unspecified, return "error: foreign socket unspecified"; if active and the foreign socket is specified, issue a SYN segment. An initial send sequence number (ISS) is selected. A SYN segment of the form <SEQ=ISS><CTL=SYN> is sent. Set SND.UNA to ISS, SND.NXT to ISS+1, enter SYN-SENT state, and return."</p> <p>https://tools.ietf.org/html/rfc793 (emphasis added)</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 15 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured such that the TCP-variant connection is at least partially closed without any keep-alive packet caused to be communicated before the TCP-variant connection is at least partially closed.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that the TCP-variant connection (e.g., the QUIC connection, etc.) is at least partially closed without any keep-alive packet caused to be communicated before the TCP-variant connection is at least partially closed.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max idle timeout transport parameter (see Section 10). . ."

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

"5. Connection Closure

Once established, an HTTP/3 connection can be used for many requests and responses over time until the connection is closed. Connection closure can happen in any of several different ways.

5.1. Idle Connections

Each QUIC endpoint declares an idle timeout during the handshake. If the connection remains idle (no packets received) for longer than this duration, the peer will assume that the connection has been closed. HTTP/3 implementations will need to open a new connection for new requests if the existing connection has been idle for longer than the server's advertised idle timeout, and SHOULD do so if approaching the idle timeout.

HTTP clients are expected to request that the transport keep connections open while there are responses outstanding for requests or server pushes, as described in Section 10.2.2 of [QUIC-TRANSPORT]. If the client is not expecting a response from the server, allowing an idle connection to time out is preferred over expending effort maintaining a connection that might not be needed. A gateway MAY maintain connections in anticipation of need rather than incur the latency cost of connection establishment to servers. Servers SHOULD NOT actively keep connections open." <https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

"3.3. Session resumption versus Keep-alive

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p> <ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods. • Resume the session after a long idle period, using 0-RTT resumption when appropriate. <p>The first strategy is the easiest, but it only applies to certain applications. Either the server or the client in a QUIC application can send PING frames as keep-alives, to prevent the connection and any on-path state from timing out.</p> <p><u>Recommendations for the use of keep-alives are application specific, mainly depending on the latency requirements and message frequency of the application. In this case, the application mapping must specify whether the client or server is responsible for keeping the application alive.</u> Note that sending PING frames more frequently than every 30 seconds over long idle periods may result in a too much unproductive traffic and power usage for some situations.</p> <p><u>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic.</u> In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with restarting the connection. This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.</p> <p>The tradeoffs between resumption and keepalive need to be evaluated on a per-application basis. However, in general applications should use keepalives only in circumstances where continued</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,742,774

	communication is highly likely; [QUIC-HTTP], for instance, recommends using PING frames for keepalive only when a request is outstanding.” https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 16 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured such that a keep-alive attribute, that is separate from the timeout attribute, is utilized for controlling communication of one or more keep-alive packets before an expiration of the time period.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that a keep-alive attribute, that is separate from the timeout attribute, is utilized for controlling communication of one or more keep-alive packets before an expiration of the time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, infringement occurs, for example, at least during a packet communication scenario when a keep-alive packet starts a detected idle time period and no other keep-alives are communicated during the idle period.</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</u></p> <p>“The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their <u>peers are still alive or to check reachability to the peer.</u> <u>The PING frame contains no additional fields.</u></p> <p>The <u>receiver of a PING frame simply needs to acknowledge the packet</u> containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,742,774

	<p>interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 17 Elements	Applicability
<p>The apparatus of claim 16 wherein the apparatus is configured such that the keep-alive attribute is calculated based on the time period.</p>	<p>Google infringes claim 16 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that the keep-alive attribute is calculated based on the time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-27</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 18 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured such that the timeout attribute is used to at least partially close the TCP-variant connection, by the TCP-variant connection being at least partially closed based on the timeout attribute.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that the timeout attribute is used to at least partially close the TCP-variant connection (e.g., the QUIC connection, etc.), by the TCP-variant connection being at least partially closed based on the timeout attribute.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p><u>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive.</u> If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"5. Connection Closure</p> <p>Once established, an HTTP/3 connection can be used for many requests and responses over time until the connection is closed. <u>Connection closure can happen in any of several different ways.</u></p> <p>5.1. Idle Connections</p> <p><u>Each QUIC endpoint declares an idle timeout during the handshake. If the connection remains idle (no packets received) for longer than this duration, the peer will assume that the connection has been closed.</u> HTTP/3 implementations will need to open a new connection for new requests if the existing connection has been idle for longer than the server's advertised idle timeout, and SHOULD do so if approaching the idle timeout.</p> <p>HTTP clients are expected to request that the transport keep connections open while there are responses outstanding for requests or server pushes, as described in Section 10.2.2 of [QUIC-TRANSPORT]. <u>If the client is not expecting a response from the server, allowing an idle connection to time out is preferred over expending effort maintaining a connection that might not be needed.</u> A gateway MAY maintain connections in anticipation of need rather than incur the latency cost of connection establishment to servers. <u>Servers SHOULD NOT actively keep connections open.</u>" https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 19 Elements	Applicability
<p>The apparatus of claim 18 wherein the apparatus is configured such that the TCP-variant connection is at least partially closed, by the second node, without sending any packet for delivery to the first node during the time period.</p>	<p>Google infringes claim 18 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that the TCP-variant connection (e.g., the QUIC connection, etc.) is at least partially closed, by the second node (e.g., the other one of the QUIC-compliant server or client, etc.), without sending any packet for delivery to the first node (e.g., one of the QUIC-compliant server or client, etc.) during the time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p><u>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive.</u> If a peer could time out</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 32 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured such that at least an acknowledgment timeout is based on the metadata.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that at least an acknowledgment timeout (e.g., idle timeout, etc.) is based on the metadata.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“19.2. PING Frame Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The <u>receiver of a PING frame simply needs to acknowledge the packet containing this frame.</u></p> <p>The <u>PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] <u>recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</u>”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 36 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured such that: the timeout attribute associated with the TCP-variant connection is capable of being used by the second node to unilaterally at least partially close the TCP-variant connection, without sending any packet for delivery to the first node during nor after the time period, by the TCP-variant connection being at least partially closed based on the timeout attribute and at least one of a formula, an expression, a function, or a policy;</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that: the timeout attribute associated with the TCP-variant connection is capable of being used by the second node to unilaterally at least partially close the TCP-variant connection (e.g., the QUIC connection, etc.), without sending any packet for delivery to the first node during nor after the time period, by the TCP-variant connection being at least partially closed based on the timeout attribute and at least one of a formula, an expression, a function, or a policy (e.g., based on the idle_timeout duration, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily determined based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p><u>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive.</u> If a peer could time out within a</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"5. Connection Closure</p> <p>Once established, an HTTP/3 connection can be used for many requests and responses over time until the connection is closed. <u>Connection closure can happen in any of several different ways.</u></p>
--	--

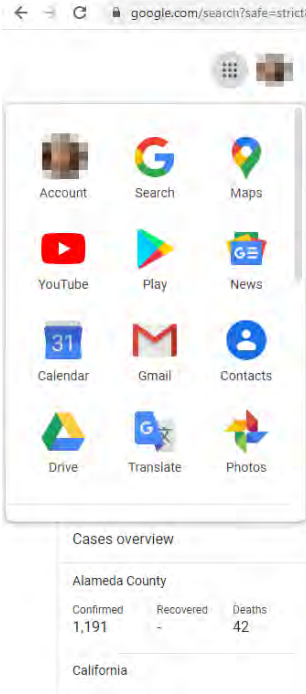
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>5.1. Idle Connections</p> <p><u>Each QUIC endpoint declares an idle timeout during the handshake. If the connection remains idle (no packets received) for longer than this duration, the peer will assume that the connection has been closed.</u> HTTP/3 implementations will need to open a new connection for new requests if the existing connection has been idle for longer than the server's advertised idle timeout, and SHOULD do so if approaching the idle timeout.</p> <p>HTTP clients are expected to request that the transport keep connections open while there are responses outstanding for requests or server pushes, as described in Section 10.2.2 of [QUIC-TRANSPORT]. <u>If the client is not expecting a response from the server, allowing an idle connection to time out is preferred over expending effort maintaining a connection that might not be needed.</u> A gateway MAY maintain connections in anticipation of need rather than incur the latency cost of connection establishment to servers. <u>Servers SHOULD NOT actively keep connections open.</u>" https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
the TCP-variant packet is not a synchronize (SYN) packet; and	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the TCP-variant packet (e.g., QUIC negotiation packet, etc.) is not a synchronize (SYN) packet.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, synchronization is required between the two sides prior to establishing a TCP-variant connection (e.g., a QUIC connection, etc.).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>7.2. Version Negotiation</p> <p>QUIC's connection establishment begins with version negotiation, since all communication between the endpoints, including packet and frame formats, relies on the two endpoints agreeing on a version.</p> <p>A QUIC connection begins with a client sending a Client Initial packet (Section 5.4.1). The details of the handshake mechanisms are described in Section 7.3, but all of the initial packets sent from the client to the server MUST use the long header format - which includes the version of the protocol being used - and they MUST be padded to at least 1200 octets.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>the TCP-variant connection is not a standard TCP connection.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the TCP-variant connection (e.g., QUIC connection, etc.) is not a standard TCP connection.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

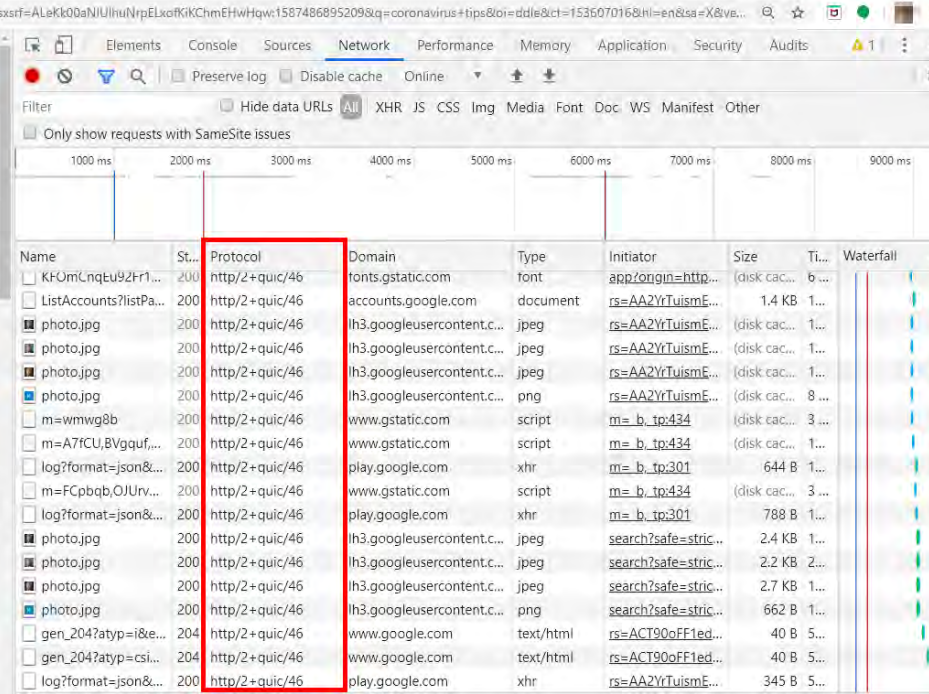


Cases overview

Alameda County

Confirmed	Recovered	Deaths
1,191	-	42

California



Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfOmCnqEu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app/origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_to:434	(disk cac...	3...	
m=A7fCU,8Vgquf,...	200	http/2+quic/46	www.gstatic.com	script	m= b_to:434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_to:301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m= b_to:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_to:301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, QUIC is a “variant” of TCP.

1. Introduction

QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.

QUIC implements techniques learned from experience with TCP, SCTP and

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,742,774

	"On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations." https://www.chromium.org/quic
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 37 Elements	Applicability
<p>The apparatus of claim 36 wherein the apparatus is configured such that: the time period starts in response to the second node sending a keep-alive packet in another data stream of the TCP-variant connection for routing towards the first node; and</p>	<p>Google infringes claim 36 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that: the time period starts in response to the second node (e.g., the other one of the QUIC-compliant server or client, etc.) sending a keep-alive packet in another data stream of the TCP-variant connection for routing towards the first node (e.g., one of the QUIC-compliant server or client, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, infringement occurs, for example, at least during a packet communication scenario when a keep-alive packet starts a detected idle time period and no other keep-alives are communicated during the idle period.</p> <p>Note: Based upon information and belief, the “second node” term is infringed by at least a server, and the “first node” term is infringed by at least a client.</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>“10.2. Idle Timeout . . . An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter</u> (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
<p>during the time period, another keep-alive packet is sent from the second node in the another data stream of the TCP-variant connection for routing towards the first node, without restarting the time period.</p>	<p>Google infringes claim 36 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein, during the time period, another keep-alive packet is sent from the second node (e.g., the other one of the QUIC-compliant server or client, etc.) in the another data stream of the TCP-variant connection (e.g., the QUIC connection, etc.) for routing towards the first node (e.g., one of the QUIC-compliant server or client, etc.), without restarting the time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p> <p>...</p> <p><u>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet.</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 38 Elements	Applicability
<p>The apparatus of claim 37 wherein the apparatus is configured such that: a user timeout attribute, that is separate from the timeout attribute, is utilized for monitoring communication of one or more acknowledgment packets, where the user timeout attribute is determined based on the time period.</p>	<p>Google infringes claim 37 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that: a user timeout attribute, that is separate from the timeout attribute, is utilized for monitoring communication of one or more acknowledgment packets, where the user timeout attribute is determined based on the time period.</p> <p>Note: Based upon information and belief, the user timeout in QUIC is based on the max_ack_delay transport parameter negotiated during setup. It is utilized to determine a probe timeout (PTO). Note that the max_ack_delay parameter is negotiated during setup. See QUIC transport draft section 13.2.1 at Hypertext Transfer Protocol Version 3 (HTTP/3) (ietf.org), section 4.7 in the IETF QUIC recovery draft at https://tools.ietf.org/html/draft-ietf-quic-recovery-34#section-4.7, and section 10.1 in the IETF QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.</p> <p>Note: Based upon information and belief, the user timeout attribute in QUIC is based on the max_ack_delay transport parameter. See section 4.7 in the IETF QUIC recovery draft at https://tools.ietf.org/html/draft-ietf-quic-recovery-34#section-4.7. The definition of max_ack_delay in the QUIC transport establishes that the user timeout is used to control ACK packets (copied below). See also section 13.2.1 "Sending ACK Frames" in the IETF QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-13.2.1.</p> <p>"max_ack_delay (0x0b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid." https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-18.2</p> <p>"13.2.1. Sending ACK Frames</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>Every packet SHOULD be acknowledged at least once, and ack-eliciting packets MUST be acknowledged at least once within the maximum ack delay. An endpoint communicates its maximum delay using the max_ack_delay transport parameter; see Section 18.2. max_ack_delay declares an explicit contract: an endpoint promises to never intentionally delay acknowledgments of an ack-eliciting packet by more than the indicated value. If it does, any excess accrues to the RTT estimate and could result in spurious or delayed retransmissions from the peer. For Initial and Handshake packets, a max_ack_delay of 0 is used. The sender uses the receiver's "max_ack_delay" value in determining timeouts for timer-based retransmission, as detailed in Section 5.2.1 of [QUIC-RECOVERY]."</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"Case 2: TCP receives a FIN from the network</p> <p>If an unsolicited FIN arrives from the network, the receiving TCP can ACK it and tell the user that the connection is closing. The user will respond with a CLOSE, upon which the TCP can send a FIN to the other TCP after sending any remaining data. <u>The TCP then waits until its own FIN is acknowledged whereupon it deletes the connection. If an ACK is not forthcoming, after the user timeout the connection is aborted and the user is told.</u></p> <p>...</p> <p>3.9. <u>Event Processing</u></p> <p>The processing depicted in this section is an example of one possible implementation. Other implementations may have slightly different processing sequences, but they should differ from those in this section only in detail, not in substance.</p> <p><u>The activity of the TCP can be characterized as responding to events.</u> The events that occur can be cast into three categories: user calls, arriving segments, and <u>timeouts</u>. This section describes the processing the TCP does in response to each of the events. In many cases the processing required depends on the state of the connection.</p> <p>...</p> <p>Timeouts</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p><u>USER TIMEOUT</u> RETRANSMISSION TIMEOUT TIME-WAIT TIMEOUT</p> <p>...</p> <p>OPEN Call</p> <p>CLOSED STATE (i.e., TCB does not exist)</p> <p><u>Create a new transmission control block (TCB) to hold connection state information. Fill in local socket identifier, foreign socket, precedence, security/compartments, and user timeout information.</u> Note that some parts of the foreign socket may be unspecified in a passive OPEN and are to be filled in by the parameters of the incoming SYN segment. Verify the security and precedence requested are allowed for this user, if not return "error: precedence not allowed" or "error: security/compartments not allowed." If passive enter the LISTEN state and return. If active and the foreign socket is unspecified, return "error: foreign socket unspecified"; if active and the foreign socket is specified, issue a SYN segment. An initial send sequence number (ISS) is selected. A SYN segment of the form <SEQ=ISS><CTL=SYN> is sent. Set SND.UNA to ISS, SND.NXT to ISS+1, enter SYN-SENT state, and return."</p> <p>https://tools.ietf.org/html/rfc793 (emphasis added)</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 39 Elements	Applicability
<p>The apparatus of claim 37 wherein the apparatus is configured such that: a keep-alive attribute, that is separate from the timeout attribute, is utilized for controlling communication of one or more keep-alive packets before an expiration of the time period, where the keep-alive attribute is determined based on the time period.</p>	<p>Google infringes claim 37 and uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that: a keep-alive attribute, that is separate from the timeout attribute, is utilized for controlling communication of one or more keep-alive packets before an expiration of the time period, where the keep-alive attribute is determined based on the time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it <u>remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO)."</p> <p>"The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>"19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their <u>peers are still alive or to check reachability to the peer.</u> <u>The PING frame contains no additional fields.</u></p> <p>The <u>receiver of a PING frame simply needs to acknowledge the packet</u> containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle timeout transport parameter</u> (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows."</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,742,774

	<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 42 Elements	Applicability
<p>A method comprising: using at least a portion of a first node: receiving information for use in: detecting a time period that results in a non-TCP connection being subject to at least partial deactivation, and sending a non-TCP packet that is based on the information and that includes a time period parameter field identifying metadata; and</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method comprising: using at least a portion of a first node (e.g., one of a QUIC-compliant server or client, etc.): receiving information for use in: detecting a time period that results in a non-TCP connection (e.g., QUIC connection, etc.) being subject to at least partial deactivation (e.g., terminating the connection, etc.), and sending a non-TCP packet that is based on the information and that includes a time period parameter field identifying metadata (e.g., a value, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>[W]e’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-00</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

(<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28>), QUIC_VERSION_IETF_DRAFT_29 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29>), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (<https://tools.ietf.org/html/draft-ietf-quic-invariants-06>) and/or IETF_QUIC_TRANSPORT_17 (<https://tools.ietf.org/html/draft-ietf-quic-transport-17>), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.

Note: Below is a web page of Google (<https://www.google.com/>).

Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KFOmCnqEu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app?origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
m=A7FCU,BVgqf...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, QUIC is a “variant” of TCP.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>1. Introduction</p> <p>QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.</p> <p>QUIC implements techniques learned from experience with TCP, SCTP and</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO)."</p> <p>"The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>sending, to a second node and for setting up the non-TCP connection, the non-TCP packet to provide the metadata to the second node, for use by the second node in determining a timeout attribute associated with the non-TCP connection.</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method including sending, to a second node (e.g., the other one of the QUIC-compliant server or client, etc.) and for setting up the non-TCP connection (e.g., QUIC connection, etc.), the non-TCP packet to provide the metadata (e.g., the value, etc.) to the second node, for use by the second node in determining a timeout attribute associated with the non-TCP connection.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>"7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 43 Elements	Applicability
<p>The method of claim 42 wherein: the determination of the timeout attribute results from a negotiation with the second node;</p>	<p>Google infringes claim 42 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein: the determination of the timeout attribute results from a negotiation with the second node (e.g., the other one of the QUIC-compliant server or client, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

“18. Transport Parameter Encoding

The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].

```
enum {
    original_connection_id(0),
    idle_timeout(1),
    stateless_reset_token(2),
    max_packet_size(3),
    initial_max_data(4),
    initial_max_stream_data_bidi_local(5),
    initial_max_stream_data_bidi_remote(6),
    initial_max_stream_data_uni(7),
    initial_max_streams_bidi(8),
    initial_max_streams_uni(9),
    ack_delay_exponent(10),
    max_ack_delay(11),
    disable_migration(12),
    preferred_address(13),
    active_connection_id_limit(14),
    (65535)
} TransportParameterId;
```

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.

“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
<p>during the time period, no packet is received from the second node in a data stream of the non-TCP connection to keep the non-TCP connection active; and</p>	<p>Google infringes claim 42 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein, during the time period, no packet is received from the second node (e.g., the other one of the QUIC-compliant server or client, etc.) in a data stream of the non-TCP connection (e.g., QUIC connection, etc.) to keep the non-TCP connection active.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“19.2. PING Frame <u>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer.</u> The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

<p>the non-TCP packet is not a synchronize (SYN) packet.</p>	<p>Google infringes claim 42 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the non-TCP packet (e.g., QUIC negotiation packet, etc.) is not a synchronize (SYN) packet.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, synchronization is required between the two sides prior to establishing a TCP-variant connection (e.g., a QUIC connection, etc.).</p> <p>7.2. Version Negotiation</p> <p>QUIC's connection establishment begins with version negotiation, since all communication between the endpoints, including packet and frame formats, relies on the two endpoints agreeing on a version.</p> <p>A QUIC connection begins with a client sending a Client Initial packet (Section 5.4.1). The details of the handshake mechanisms are described in Section 7.3, but all of the initial packets sent from the client to the server MUST use the long header format - which includes the version of the protocol being used - and they MUST be padded to at least 1200 octets.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p>QUIC: A UDP-Based Multiplexed and Secure Transport https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 44 Elements	Applicability
<p>The method of claim 43 wherein: the timeout attribute associated with the non-TCP connection is capable of being used to unilaterally cause the non-TCP connection to at least partially close, based on the timeout attribute and at least one of a formula, an expression, a function, or a policy;</p>	<p>Google infringes claim 43 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein: the timeout attribute associated with the non-TCP connection (e.g., the QUIC connection, etc.) is capable of being used to unilaterally cause the non-TCP connection to at least partially close, based on the timeout attribute and at least one of a formula, an expression, a function, or a policy (e.g., based on the idle_timeout duration, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, the “second node” term is infringed by at least a server, and the “first node” term is infringed by at least a client.</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily determined based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p><u>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive.</u> If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max idle timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"5. Connection Closure</p> <p>Once established, an HTTP/3 connection can be used for many requests and responses over time until the connection is closed. <u>Connection closure can happen in any of several different ways.</u></p> <p>5.1. Idle Connections</p> <p><u>Each QUIC endpoint declares an idle timeout during the handshake. If the connection remains idle (no packets received) for longer than this duration, the peer will assume that the connection has been closed.</u> HTTP/3 implementations will need to open a new connection for new requests if the existing connection has been idle for longer than the server's advertised idle timeout, and SHOULD do so if approaching the idle timeout.</p> <p>HTTP clients are expected to request that the transport keep connections open while there are responses outstanding for requests or server pushes, as described in Section 10.2.2 of [QUIC-TRANSPORT]. <u>If the client is not expecting a response from the server, allowing an idle connection to time out is preferred over expending effort maintaining a connection that might not be needed.</u> A gateway MAY maintain connections in anticipation of need rather than incur the latency cost of connection establishment to servers. <u>Servers SHOULD NOT actively keep connections open.</u>" https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
the time period starts in response to the second node sending a keep-alive packet in the non-TCP connection for routing towards the first node; and	<p>Google infringes claim 43 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the time period starts in response to the second node (e.g., the other one of the QUIC-compliant server or client, etc.) sending a keep-alive packet in the non-TCP connection (e.g., QUIC connection, etc.) for routing towards the first node (e.g., one of the QUIC-compliant server or client, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>"10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>...</p> <p><u>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4."</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>"19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</u></p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max idle timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
during the time period, another keep-alive packet is sent in the non-TCP	<p>Google infringes claim 43 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein, during the</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

<p>connection for routing towards the first node, without restarting the time period.</p>	<p>time period, another keep-alive packet is sent in the non-TCP connection (e.g., the QUIC connection, etc.) for routing towards the first node (e.g., one of the QUIC-compliant server or client, etc.), without restarting the time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout . . . <u>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet.</u> Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.” QUIC: A UDP-Based Multiplexed and Secure Transport https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“19.2. PING Frame <u>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer.</u> The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout</p>
---	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,742,774

	interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 51 Elements	Applicability
<p>The method of claim 43 wherein the timeout attribute associated with the non-TCP connection is used by the first node to unilaterally at least partially close the non-TCP connection.</p>	<p>Google infringes claim 43 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the timeout attribute associated with the non-TCP connection is used by the first node (e.g., one of the QUIC-compliant server or client, etc.) to unilaterally at least partially close the non-TCP connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"5. Connection Closure</p> <p>Once established, an HTTP/3 connection can be used for many requests and responses over time until the connection is closed. <u>Connection closure can happen in any of several different ways.</u></p> <p>5.1. Idle Connections</p> <p><u>Each QUIC endpoint declares an idle timeout during the handshake. If the connection remains idle (no packets received) for longer than this duration, the peer will assume that the connection has been closed.</u> HTTP/3 implementations will need to open a new connection for new requests if the existing connection has been idle for longer than the server's advertised idle timeout, and SHOULD do so if approaching the idle timeout.</p> <p>HTTP clients are expected to request that the transport keep connections open while there are responses outstanding for requests or server pushes, as described in Section 10.2.2 of [QUIC-</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,742,774

	<p>TRANSPORT]. <u>If the client is not expecting a response from the server, allowing an idle connection to time out is preferred over expending effort maintaining a connection that might not be needed.</u> A gateway MAY maintain connections in anticipation of need rather than incur the latency cost of connection establishment to servers. <u>Servers SHOULD NOT actively keep connections open.</u>" https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 59 Elements	Applicability
<p>The method of claim 43 wherein a user timeout attribute, that is separate from the timeout attribute, is utilized for monitoring communication of one or more acknowledgment packets, where the user timeout attribute is determined based on the time period.</p>	<p>Google infringes claim 43 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein a user timeout attribute, that is separate from the timeout attribute, is utilized for monitoring communication of one or more acknowledgment packets, where the user timeout attribute is determined based on the time period.</p> <p>Note: Based upon information and belief, the user timeout in QUIC is based on the max_ack_delay transport parameter negotiated during setup. It is utilized to determine a probe timeout (PTO). Note that the max_ack_delay parameter is negotiated during setup. See QUIC transport draft section 13.2.1 at Hypertext Transfer Protocol Version 3 (HTTP/3) (ietf.org), section 4.7 in the IETF QUIC recovery draft at https://tools.ietf.org/html/draft-ietf-quic-recovery-34#section-4.7, and section 10.1 in the IETF QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.</p> <p>Note: Based upon information and belief, the user timeout attribute in QUIC is based on the max_ack_delay transport parameter. See section 4.7 in the IETF QUIC recovery draft at https://tools.ietf.org/html/draft-ietf-quic-recovery-34#section-4.7. The definition of max_ack_delay in the QUIC transport establishes that the user timeout is used to control ACK packets (copied below). See also section 13.2.1 "Sending ACK Frames" in the IETF QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-13.2.1.</p> <p>"max_ack_delay (0x0b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid." https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-18.2</p> <p>"13.2.1. Sending ACK Frames</p> <p>Every packet SHOULD be acknowledged at least once, and ack-eliciting packets MUST be acknowledged at least once within the maximum ack delay. An endpoint communicates its</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>maximum delay using the max_ack_delay transport parameter; see Section 18.2. max_ack_delay declares an explicit contract: an endpoint promises to never intentionally delay acknowledgments of an ack-eliciting packet by more than the indicated value. If it does, any excess accrues to the RTT estimate and could result in spurious or delayed retransmissions from the peer. For Initial and Handshake packets, a max_ack_delay of 0 is used. The sender uses the receiver's "max_ack_delay" value in determining timeouts for timer-based retransmission, as detailed in Section 5.2.1 of [QUIC-RECOVERY]."</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"Case 2: TCP receives a FIN from the network</p> <p>If an unsolicited FIN arrives from the network, the receiving TCP can ACK it and tell the user that the connection is closing. The user will respond with a CLOSE, upon which the TCP can send a FIN to the other TCP after sending any remaining data. <u>The TCP then waits until its own FIN is acknowledged whereupon it deletes the connection. If an ACK is not forthcoming, after the user timeout the connection is aborted and the user is told.</u></p> <p>...</p> <p>3.9. <u>Event Processing</u></p> <p>The processing depicted in this section is an example of one possible implementation. Other implementations may have slightly different processing sequences, but they should differ from those in this section only in detail, not in substance.</p> <p><u>The activity of the TCP can be characterized as responding to events.</u> The events that occur can be cast into three categories: user calls, arriving segments, and <u>timeouts</u>. This section describes the processing the TCP does in response to each of the events. In many cases the processing required depends on the state of the connection.</p> <p>...</p> <p>Timeouts</p> <p><u>USER TIMEOUT</u></p> <p><u>RETRANSMISSION TIMEOUT</u></p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,742,774

	<p>TIME-WAIT TIMEOUT</p> <p>...</p> <p>OPEN Call</p> <p>CLOSED STATE (i.e., TCB does not exist)</p> <p><u>Create a new transmission control block (TCB) to hold connection state information. Fill in local socket identifier, foreign socket, precedence, security/compartments, and user timeout information.</u> Note that some parts of the foreign socket may be unspecified in a passive OPEN and are to be filled in by the parameters of the incoming SYN segment. Verify the security and precedence requested are allowed for this user, if not return "error: precedence not allowed" or "error: security/compartments not allowed." If passive enter the LISTEN state and return. If active and the foreign socket is unspecified, return "error: foreign socket unspecified"; if active and the foreign socket is specified, issue a SYN segment. An initial send sequence number (ISS) is selected. A SYN segment of the form <SEQ=ISS><CTL=SYN> is sent. Set SND.UNA to ISS, SND.NXT to ISS+1, enter SYN-SENT state, and return."</p> <p>https://tools.ietf.org/html/rfc793 (emphasis added)</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 65 Elements	Applicability
<p>The method of claim 43 wherein the timeout attribute associated with the non-TCP connection is capable of being used by the first node to unilaterally at least partially close the non-TCP connection, without sending to the second node any packet that is related to the detection of the time period.</p>	<p>Google infringes claim 43 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the timeout attribute associated with the non-TCP connection (e.g., the QUIC connection, etc.) is capable of being used by the first node (e.g., one of the QUIC-compliant server or client, etc.) to unilaterally at least partially close the non-TCP connection, without sending to the second node (e.g., the other one of the QUIC-compliant server or client, etc.) any packet that is related to the detection of the time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p>

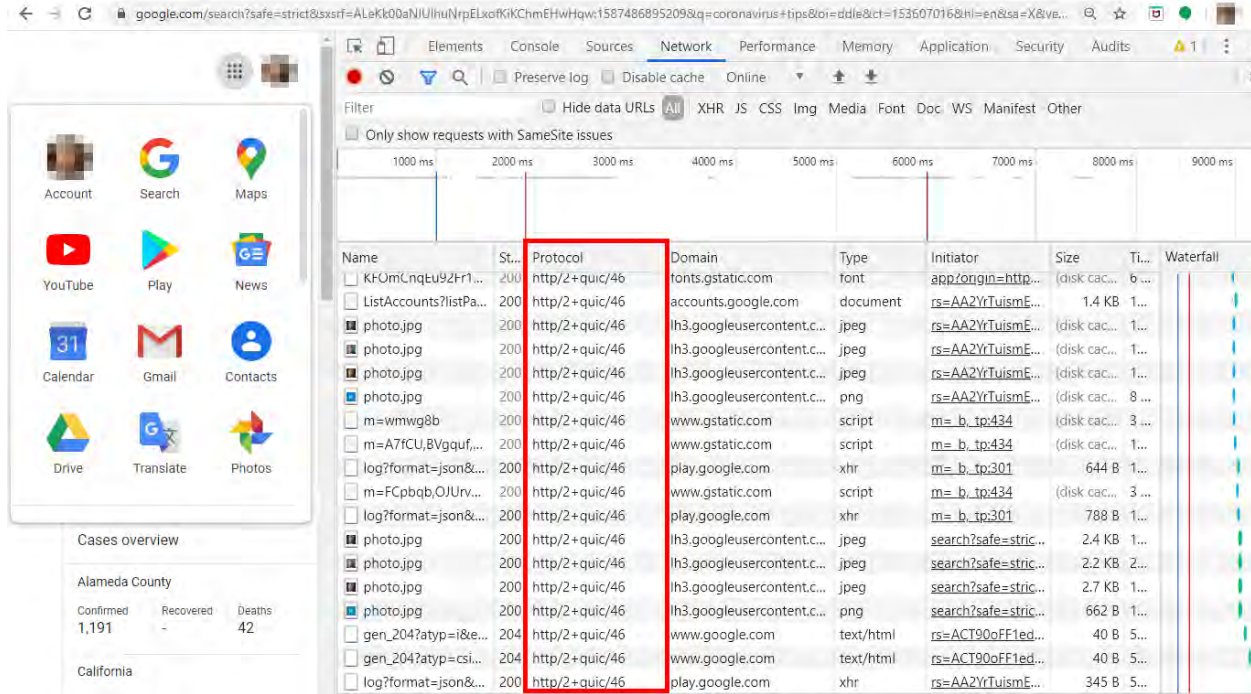
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"5. Connection Closure</p> <p>Once established, an HTTP/3 connection can be used for many requests and responses over time until the connection is closed. <u>Connection closure can happen in any of several different ways.</u></p> <p>5.1. Idle Connections</p> <p><u>Each QUIC endpoint declares an idle timeout during the handshake. If the connection remains idle (no packets received) for longer than this duration, the peer will assume that the connection has been closed.</u> HTTP/3 implementations will need to open a new connection for new requests if the existing connection has been idle for longer than the server's advertised idle timeout, and SHOULD do so if approaching the idle timeout.</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,742,774

	<p>HTTP clients are expected to request that the transport keep connections open while there are responses outstanding for requests or server pushes, as described in Section 10.2.2 of [QUIC-TRANSPORT]. <u>If the client is not expecting a response from the server, allowing an idle connection to time out is preferred over expending effort maintaining a connection that might not be needed.</u> A gateway MAY maintain connections in anticipation of need rather than incur the latency cost of connection establishment to servers. <u>Servers SHOULD NOT actively keep connections open.</u>" https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 67 Elements	Applicability
<p>The method of claim 43 wherein the non-TCP connection is not a TCP extension.</p>	<p>Google infringes claim 43 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the non-TCP connection (e.g., QUIC connection, etc.) is not a TCP extension.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p>  <p>Note: As set forth below, QUIC is a “variant” of TCP.</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,742,774

	<p>1. Introduction</p> <p>QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.</p> <p>QUIC implements techniques learned from experience with TCP, SCTP and</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

Claim 72 Elements	Applicability
<p>The method of claim 43 wherein the time period is not a user timeout period.</p>	<p>Google infringes claim 43 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the time period is not a user timeout period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value (time period) in seconds that is encoded as an unsigned 16-bit integer, from which the user timeout period is separate.</p> <p>idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</p> <p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>"3.9. Event Processing</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,742,774

	<p>The processing depicted in this section is an example of one possible implementation. Other implementations may have slightly different processing sequences, but they should differ from those in this section only in detail, not in substance.</p> <p><u>The activity of the TCP can be characterized as responding to events.</u> The events that occur can be cast into three categories: user calls, arriving segments, and <u>timeouts</u>. This section describes the processing the TCP does in response to each of the events. In many cases the processing required depends on the state of the connection.</p> <p>...</p> <p>Timeouts</p> <p><u>USER TIMEOUT</u> <u>RETRANSMISSION TIMEOUT</u> <u>TIME-WAIT TIMEOUT</u></p> <p>...</p> <p>OPEN Call</p> <p>CLOSED STATE (i.e., TCB does not exist)</p> <p><u>Create a new transmission control block (TCB) to hold connection state information. Fill in local socket identifier, foreign socket, precedence, security/compartments, and user timeout information.</u> Note that some parts of the foreign socket may be unspecified in a passive OPEN and are to be filled in by the parameters of the incoming SYN segment. Verify the security and precedence requested are allowed for this user, if not return "error: precedence not allowed" or "error: security/compartments not allowed." If passive enter the LISTEN state and return. If active and the foreign socket is unspecified, return "error: foreign socket unspecified"; if active and the foreign socket is specified, issue a SYN segment. An initial send sequence number (ISS) is selected. A SYN segment of the form <SEQ=ISS><CTL=SYN> is sent. Set SND.UNA to ISS, SND.NXT to ISS+1, enter SYN-SENT state, and return."</p> <p>https://tools.ietf.org/html/rfc793 (emphasis added)</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,742,774

Caveat: The notes and/or cited excerpts utilized herein are set forth for illustrative purposes only and are not meant to be limiting in any manner. For example, the notes and/or cited excerpts, may or may not be supplemented or substituted with different excerpt(s) of the relevant reference(s), as appropriate. Further, to the extent any error(s) and/or omission(s) exist herein, all rights are reserved to correct the same.

JENAM TECH LLC’S FIRST AMENDED SET OF INFRINGEMENT CONTENTIONS

U.S. Patent No. 10,069,945 – Google LLC Claims 7, 8, 9, 10, 11, 12, 65, 67, 68, and 69

Jenam Tech, LLC (“Jenam”) provides evidence of infringement of claims 7, 8, 9, 10, 11, 12, 65, 67, 68, and 69 of U.S. Patent No. 10,069,945 (hereinafter “the ‘945 patent”) by Google LLC (“Google”). In support thereof, Jenam provides the following claim charts.

“Accused Instrumentalities” as used herein refers to at least one or more websites or web addresses including, but not limited to www.google.com, stored and/or hosted on one or more servers owned or under the control of Google. For the sake of clarity, Jenam alleges that all of Google’s various websites, products, and platforms utilizing QUIC infringe the ‘945 patent, including, but not limited to Google Edge Network, Google Cloud, Chrome Enterprise, G suite, Google Play, Chrome, Android (Android Enterprise, Android Messages (RCS)), Duo, Google Ads, Adwords, Google Analytics, Youtube, Google Mobile apps, Google shopping, and Google Maps. A list of Google’s “products” can be found at https://about.google/intl/en_us/products/. On information and belief, the Accused Instrumentalities referred to in the charts below are representative of Google’s use of QUIC in Google’s other websites, products and platforms. These claim charts demonstrate Google’s infringement, and provide notice of such infringement, by comparing each element of the asserted claims to corresponding components, aspects, and/or features of the Accused Instrumentalities. These claim charts are not intended to constitute an expert report on infringement. These claim charts include information provided by way of example, and not by way of limitation.

The analysis set forth below is based only upon information from publicly available resources regarding the Infringing Instrumentalities. An analysis of Google’s (or other third parties’) technical documentation and/or software source code may assist in fully identify all infringing features and functionality. Accordingly, Jenam reserves the right to supplement this infringement analysis once such information is made available to Jenam . Furthermore, Jenam reserves the right to revise this infringement analysis, as appropriate, upon issuance of a court order construing any terms recited in the asserted claims.

Unless otherwise noted, Jenam contends that Google directly infringes the ‘945 patent in violation of 35 U.S.C. § 271(a) by selling, offering to sell, making, using, and/or importing the Infringing Instrumentalities. The following exemplary analysis demonstrates that infringement. Unless otherwise noted, Jenam further contends that the evidence below supports a finding of indirect infringement under 35 U.S.C. §§ 271(b) and/or (c), in conjunction with other evidence of liability under one or more of those subsections. Google makes, uses, sells, imports, or offers for sale in the United States, or has made, used, sold, imported, or offered for sale in the past, without authority, or induces others to make, use, sell, import, or offer for sale in the United States, or has induced others to make, use, sell, import, or offer for sale in the past, without authority products, equipment, or services that infringe claims 7, 8, 9, 10, 11, 12, 65, 67, 68, and 69 of the ‘945 patent, including without limitation, the Accused Instrumentalities.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Unless otherwise noted, Jenam believes and contends that each element of each claim asserted herein is literally met through Google's provision of the Infringing Instrumentalities. However, to the extent that Google attempts to allege that any asserted claim element is not literally met, Jenam believes and contends that such elements are met under the doctrine of equivalents. More specifically, in its investigation and analysis of the Infringing Instrumentalities, Jenam did not identify any substantial differences between the elements of the patent claims and the corresponding features of the Infringing Instrumentalities, as set forth herein. In each instance, the identified feature of the Infringing Instrumentalities performs at least substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim element.

To the extent the chart of an asserted claim relies on evidence about certain specifically-identified Accused Instrumentalities, Jenam asserts that, on information and belief, any similarly-functioning instrumentalities also infringes the charted claim. Jenam reserves the right to amend this infringement analysis based on other products made, used, sold, imported, or offered for sale by Google. Jenam also reserves the right to amend this infringement analysis by citing other claims of the '945 patent, not listed in the claim chart, that are infringed by the Accused Instrumentalities. Jenam further reserves the right to amend this infringement analysis by adding, subtracting, or otherwise modifying content in the "Accused Instrumentalities" column of each chart.

Claim 1 Elements	Applicability
A computer-implemented method, comprising: causing access to be provided to a server computer including: a non-transitory memory storing a network application, and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application to operate in accordance with a first protocol including a transmission control protocol (TCP); causing a	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method comprising: causing access to be provided to a server computer including: a non-transitory memory storing a network application (e.g., server software, etc.), and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application to operate in accordance with a first protocol including a transmission control protocol (TCP); causing a TCP connection to be established with a client computer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their "Google Cloud Platform" uses QUIC:</p> <p>"Google Cloud Platform Blog...</p>

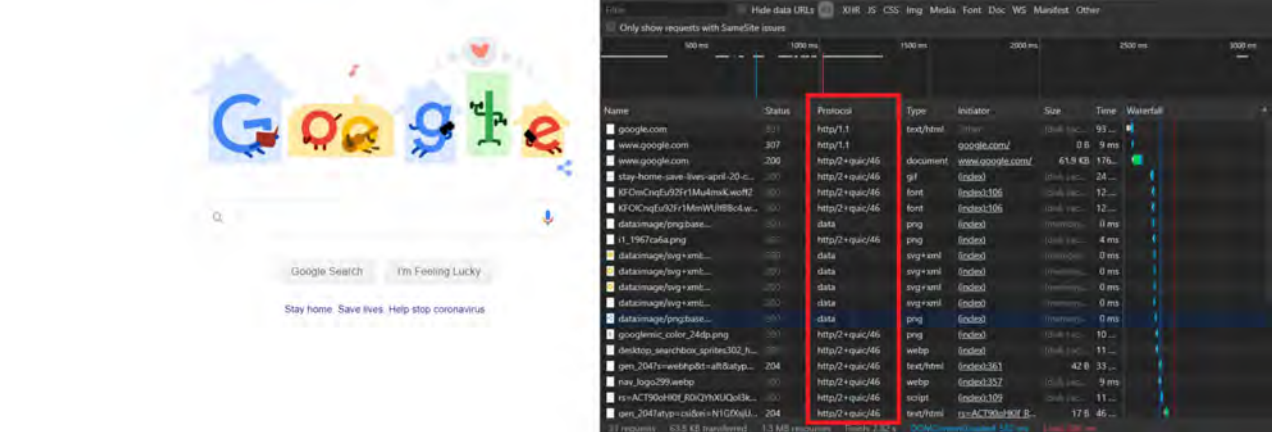
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

<p>TCP connection to be established with a client computer, by:</p>	<p>we're happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The "Google Cloud Platform" that, as established above, uses QUIC, is also used for ALL of Google's services:</p> <p>"Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube" https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13</p>
---	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> <u>Note:</u> Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities. </p> <p> <u>Note:</u> Below is a web page of Google (https://www.google.com/). </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	 <p>Note: The aforementioned web page of Google (https://www.google.com/) is configured for causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer.</p>
<p>communicating a segment including at least one first synchronize bit,</p> <p>communicating a first acknowledgement of the segment, and at least one second synchronize bit, and</p> <p>communicating a second acknowledgement;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including: a) communicating a segment including at least one first synchronize bit; b) communicating a first acknowledgement of the segment, and at least one second synchronize bit; and c) communicating a second acknowledgement.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a) is met by 1); b) is met by 2)/3), and c) is met by 4).</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p>

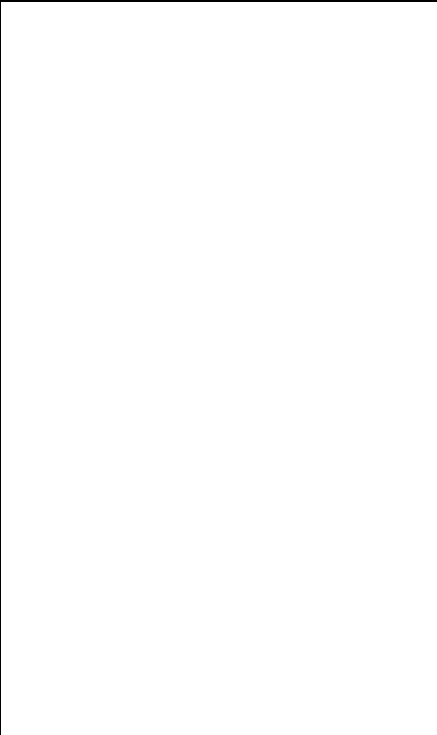
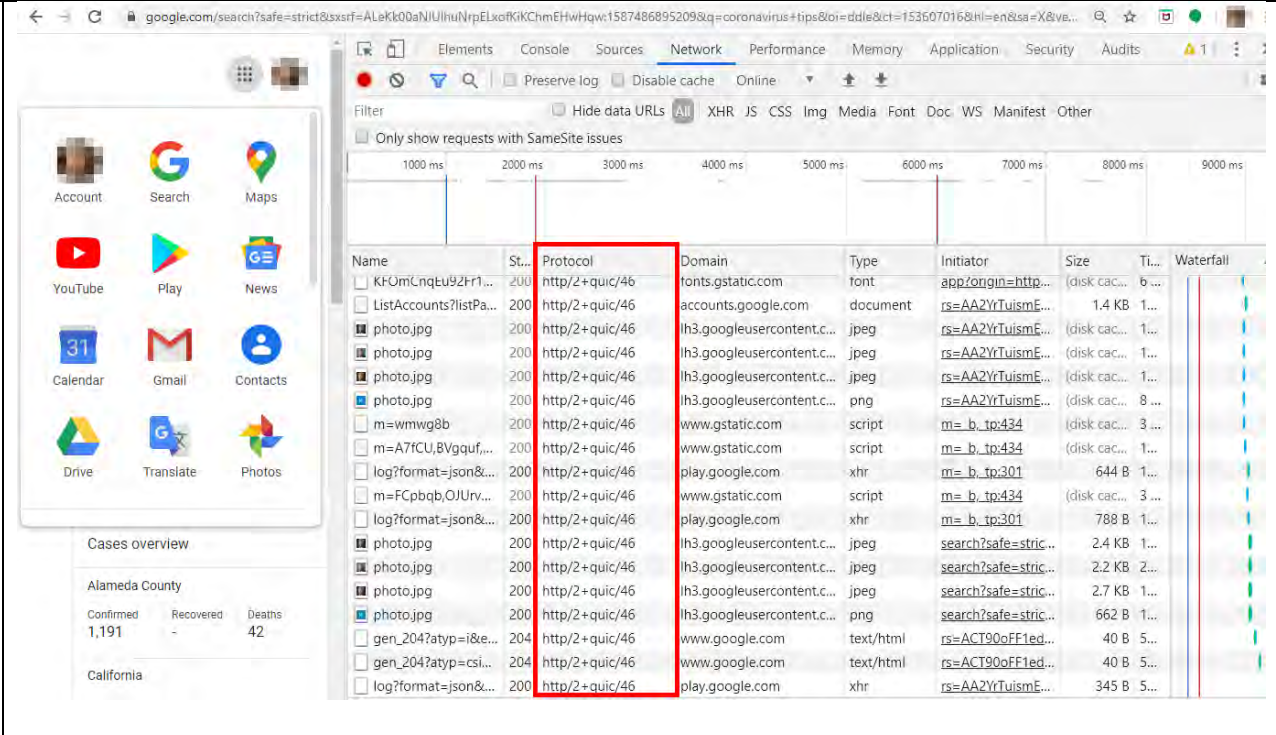
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y</p> <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3]."</p> <p>"Request for Comments" (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
<p>causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including causing first data to be communicated from the server computer to the client computer (e.g., device that receives the HTML pages, etc.) utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>"3. Stream States</p> <p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2]</u>. This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state." https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)</p>
causing the server computer to permit second data, from the user of the client computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP); and	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including causing the server computer to permit second data (e.g., data indicating navigation to an HTML page where the QUIC protocol is in use, etc.), from the user of the client computer (e.g., device that receives the HTML pages, etc.), to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	
<p>causing access to be provided, to the client computer, to code that causes the client computer to operate in accordance with a second protocol that is separate from the TCP, in order to establish a second protocol connection with another server computer, by: receiving a packet,</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including causing access to be provided, to the client computer (e.g., device that receives the HTML pages, etc.), to code that causes the client computer to operate in accordance with a second protocol (e.g., QUIC, etc.) that is separate from the TCP, in order to establish a second protocol connection (e.g., a QUIC connection, etc.) with another server (e.g., site, etc.) computer, by: receiving a packet (e.g., a handshake packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“Google AdSense provides a way for publishers to earn money from their online content. AdSense works by matching ads to your site based on your content and visitors. ... You make your ad spaces available by pasting ad code on your site, and choose where you want the ads to appear.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>https://support.google.com/adsense/answer/6242051?hl=en#zippy=%2Cwhat-is-adsense%2Cchow-does-adsense-differ-from-other-ad-networks</p> <p>Note: As set forth below, QUIC is separate from TCP.</p> <p>1. Introduction</p> <p>QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.</p> <p>QUIC implements techniques learned from experience with TCP, SCTP and</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>7.2. Version Negotiation</p> <p>QUIC’s connection establishment begins with version negotiation, since all communication between the endpoints, including packet and frame formats, relies on the two endpoints agreeing on a version.</p> <p>A QUIC connection begins with a client sending a Client Initial packet (Section 5.4.1). The details of the handshake mechanisms are described in Section 7.3, but all of the initial packets sent from the client to the server MUST use the long header format - which includes the version of the protocol being used - and they MUST be padded to at least 1200 octets.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. However, when the server generates a Version Negotiation packet, it cannot randomly generate a reserved version number. This is because the server is required to include the same value in its transport parameters (see Section 7.4.4). To avoid the selected version number</p> <ul style="list-style-type: none"> o The initial handshake packet from the client needs to fit in a single packet (#338) <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>3.1. Low-Latency Connection Establishment</p> <p>QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

detecting an idle time period parameter field in the packet,	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including detecting an idle time period parameter field (e.g., idle_timeout parameter field, etc.) in the packet (e.g., a handshake packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter that is detected by a recipient of such packet.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>The format of the transport parameters is the TransportParameters struct from Figure 6. This is described using the presentation language from Section 3 of [I-D.ietf-tls-tls13].</p> <pre>uint32 QuicVersion;</pre> <pre>enum { initial_max_stream_data(0), initial_max_data(1), initial_max_stream_id(2), idle_timeout(3), omit_connection_id(4), max_packet_size(5), stateless_reset_token(6), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>identifying metadata in the idle time period parameter field for an idle time period, where, after the idle time period is detected, the second protocol connection is deemed inactive, and</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including identifying metadata (e.g., a value, etc.) in the idle time period parameter field (e.g., idle_timeout parameter field, etc.) for an idle time period, where, after the idle time period is detected, the second protocol connection (e.g., QUIC connection, etc.) is deemed inactive.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

	<p>comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	+-----+-----+-----+		
	Value	Parameter Name	Specification
	0x0000	original_connection_id	Section 18.1
	0x0001	idle_timeout	Section 18.1
	0x0002	stateless_reset_token	Section 18.1
	0x0003	max_packet_size	Section 18.1
	0x0004	initial_max_data	Section 18.1
	0x0005	initial_max_stream_data_bidi_local	Section 18.1
	0x0006	initial_max_stream_data_bidi_remote	Section 18.1
	0x0007	initial_max_stream_data_uni	Section 18.1
	0x0008	initial_max_streams_bidi	Section 18.1
	0x0009	initial_max_streams_uni	Section 18.1
	0x000a	ack_delay_exponent	Section 18.1
	0x000b	max_ack_delay	Section 18.1
	0x000c	disable_migration	Section 18.1
	0x000d	preferred_address	Section 18.1
	0x000e	active_connection_id_limit	Section 18.1
	+-----+-----+-----+		
	Table 6: Initial QUIC Transport Parameters Entries		

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO)."</p> <p><u>"The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried."</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
creating or modifying, by the client computer and based on the metadata, a timeout attribute associated with the second protocol connection.	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including creating or modifying, by the client computer (e.g., device that receives the HTML pages, etc.) and based on the metadata (e.g., the value, etc.), a timeout attribute associated with the second protocol connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>"idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle timeout transport parameter</u> (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---


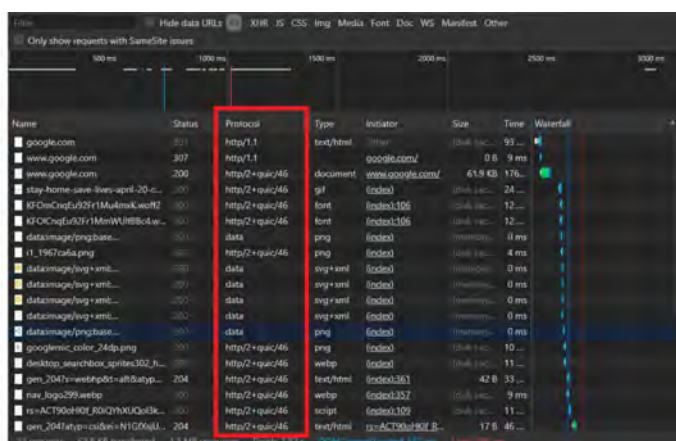
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 7 Elements	Applicability
<p>The computer-implemented method of claim 1 wherein the access is caused to be provided to the code, by permitting the code to be received from the another server computer.</p>	<p>Google infringes claim 1 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the access is caused to be provided to the code, by permitting the code to be received from the another server (e.g., site, etc.) computer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“Google AdSense provides a way for publishers to earn money from their online content. AdSense works by matching ads to your site based on your content and visitors. ... You make your ad spaces available by pasting ad code on your site, and choose where you want the ads to appear.” https://support.google.com/adsense/answer/6242051?hl=en#zippy=%2Cwhat-is-adsense%2Cchow-does-adsense-differ-from-other-ad-networks</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p> <div data-bbox="808 846 1192 1130" data-label="Image"> </div> <div data-bbox="1207 808 1890 1242" data-label="Figure"> </div> <p>Note: The aforementioned web page of Google (https://www.google.com/) is configured for causing first data to be communicated from the server computer to the client computer utilizing the TCP connection</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

	in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer.
--	---


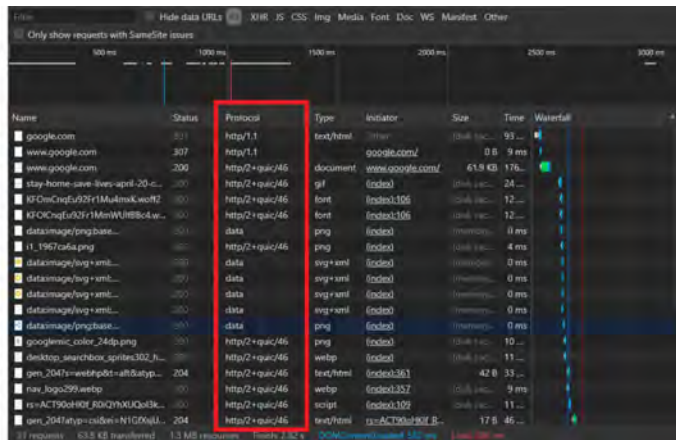
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 8 Elements	Applicability
<p>The computer-implemented method of claim 1 wherein the access is caused to be provided to the code, by permitting the code to be received from yet another server that is different from the server computer and the another server computer.</p>	<p>Google infringes claim 1 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the access is caused to be provided to the code, by permitting the code to be received from yet another server (e.g., site, etc.) that is different from the server computer and the another server (e.g., site, etc.) computer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“Google AdSense provides a way for publishers to earn money from their online content. AdSense works by matching ads to your site based on your content and visitors. ... You make your ad spaces available by pasting ad code on your site, and choose where you want the ads to appear.” https://support.google.com/adsense/answer/6242051?hl=en#zippy=%2Cwhat-is-adsense%2Cchow-does-adsense-differ-from-other-ad-networks</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p> <div style="display: flex; justify-content: space-around; align-items: center;">   </div>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

	<p>Note: The aforementioned web page of Google (https://www.google.com/) is configured for causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 9 Elements	Applicability
The computer-implemented method of claim 1 wherein the code causes the client computer to operate such that the packet is received before the second protocol connection is established.	<p>Google infringes claim 1 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the code causes the client computer (e.g., device that receives the HTML pages, etc.) to operate such that the packet (e.g., QUIC negotiation packet, etc.) is received before the second protocol connection (e.g., QUIC connection, etc.) is established.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p> <div style="display: flex; justify-content: space-around; align-items: center;">   </div> <p>Note: The aforementioned web page of Google (https://www.google.com/) is configured for causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 10 Elements	Applicability
<p>The computer-implemented method of claim 1 wherein the code causes the client computer to operate such that, after the second protocol connection is established, another packet is received with the idle time period parameter field.</p>	<p>Google infringes claim 1 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the code causes the client computer (e.g., device that receives the HTML pages, etc.) to operate such that, after the second protocol connection (e.g., QUIC connection, etc.) is established, another packet is received with the idle time period parameter field.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters</u>. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 11 Elements	Applicability
<p>The computer-implemented method of claim 1 wherein the code causes the client computer to operate such that the timeout attribute is subject to a global setting.</p>	<p>Google infringes claim 1 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the code causes the client computer (e.g., device that receives the HTML pages, etc.) to operate such that the timeout attribute is subject to a global setting.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the <code>idle_timeout</code> value is encoded as an unsigned 16-bit integer.</p> <p><code>idle_timeout (0x0003):</code> The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</p> <p>7.8. Connection Termination</p> <p>Connections should remain open until they become idle for a pre-negotiated period of time. A QUIC connection, once established, can be terminated in one of three ways:</p> <p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

	<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 12 Elements	Applicability
<p>The computer-implemented method of claim 1 wherein the code causes the client computer to operate such that the timeout attribute is subject to a connection-specific setting that is capable of overriding a global setting.</p>	<p>Google infringes claim 1 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the code causes the client computer (e.g., device that receives the HTML pages, etc.) to operate such that the timeout attribute is subject to a connection-specific setting that is capable of overriding a global setting.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p><code>idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</code></p> <p>7.8. Connection Termination</p> <p><code>Connections should remain open until they become idle for a pre-negotiated period of time. A QUIC connection, once established, can be terminated in one of three ways:</code></p> <p>7.8.2. Idle Timeout</p> <p><code>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</code></p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

	<p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 34 Elements	Applicability
<p>A computer-implemented method comprising: providing access to a server computer including: a non-transitory memory storing a network application, and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application to operate in accordance with a first protocol including a transmission control protocol (TCP); causing a TCP connection to be established with a client computer, by</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method comprising: providing access to a server computer including: a non-transitory memory storing a network application (e.g., server software, etc.), and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application to operate in accordance with a first protocol including a transmission control protocol (TCP); causing a TCP connection to be established with a client computer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>[W]e’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p>Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34>), QUIC_VERSION_IETF_DRAFT_25 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25>), QUIC_VERSION_IETF_DRAFT_27 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27>), QUIC_VERSION_IETF_DRAFT_28 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28>), QUIC_VERSION_IETF_DRAFT_29 (<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29>), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (<https://tools.ietf.org/html/draft-ietf-quic-invariants-06>) and/or IETF_QUIC_TRANSPORT_17 (<https://tools.ietf.org/html/draft-ietf-quic-transport-17>), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.

Note: Below is a web page of Google (<https://www.google.com/>).



 A screenshot of a web browser's developer tools, specifically the network tab. The table shows a list of requests made by the browser. The first request is highlighted with a red box. The table has columns for Name, Status, Protocol, Type, Initiator, Size, Time, and Waterfall.

Name	Status	Protocol	Type	Initiator	Size	Time	Waterfall
google.com	200	http/1.1	text/html	google.com	61.9 KB	93 ms	
www.google.com	307	http/1.1	document	google.com	61.9 KB	176 ms	
stay-home-save-lives-april-20-c...	200	http/2+quic/46	gif	google.com	24 ms		
KfOwCrefu02t1MukmKwR2	200	http/2+quic/46	font	font.gstatic.com	12 ms		
KfOwCrefu02t1MukmKwR2	200	http/2+quic/46	font	font.gstatic.com	12 ms		
data:image/png;base64...	200	data	png	font.gstatic.com	0 ms		
11_1967c6a.png	200	http/2+quic/46	png	font.gstatic.com	4 ms		
data:image/svg+xml...	200	data	svg+xml	font.gstatic.com	0 ms		
data:image/svg+xml...	200	data	svg+xml	font.gstatic.com	0 ms		
data:image/svg+xml...	200	data	svg+xml	font.gstatic.com	0 ms		
data:image/png;base64...	200	data	png	font.gstatic.com	0 ms		
googlemic_color_34dp.png	200	http/2+quic/46	png	font.gstatic.com	10 ms		
desktop_searchbox_sprites302_h...	200	http/2+quic/46	webp	font.gstatic.com	11 ms		
gen_204D+webpds+alt&atyp...	204	http/2+quic/46	text/html	font.gstatic.com	42 B	33 ms	
nav_logo299.webp	200	http/2+quic/46	webp	font.gstatic.com	9 ms		
rs=ACT90dH02_X0QYXUQdK...	200	http/2+quic/46	script	font.gstatic.com	11 ms		
gen_204f&atyp=cdn&=N1GdM...	204	http/2+quic/46	text/html	font.gstatic.com	17 B	46 ms	

Note: The aforementioned web page of Google (<https://www.google.com/>) is configured for causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

<p>communicating a segment including at least one first synchronize bit;</p> <p>communicating a first acknowledgement of the segment, and at least one second synchronize bit; and</p> <p>communicating a second acknowledgement;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including, a) communicating a segment including at least one first synchronize bit; b) communicating a first acknowledgement of the segment, and at least one second synchronize bit; and c) communicating a second acknowledgement.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a) is met by 1); b) is met by 2)/3), and c) is met by 4).</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <p>1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y</p> <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3].”</p> <p>“Request for Comments” (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
---	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

<p>causing first data to be communicated from the server computer to the client computer utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including causing first data to be communicated from the server computer to the client computer (e.g., device that receives the HTML pages, etc.) utilizing the TCP connection in accordance with the TCP protocol and a hypertext transfer protocol (HTTP), for being presented to a user of the client computer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“3. Stream States</p> <p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p> <p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2].</u> This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state.” https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)</p>
---	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

causing the server computer to permit second data, from the user of the client computer, to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP); and

Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including causing the server computer to permit second data (e.g., data indicating navigation to an HTML page where the QUIC protocol is in use, etc.), from the user of the client computer (e.g., device that receives the HTML pages, etc.), to be received at the server computer from the client computer utilizing the TCP connection in accordance with the TCP protocol and the hypertext transfer protocol (HTTP).

See excerpt(s) below, for example (emphasis added, if any):

Note: Below is a web page of Google (<https://www.google.com/>).

Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfOmCnqEu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app/origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	3...	
m=A7fCU,BVgquf...	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_tp301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_tp301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stic...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stic...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stic...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stic...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

<p>providing access to code that, after use by the client computer, results in the client computer operating in accordance with a second protocol that is separate from the TCP, in order to establish a second protocol connection with another server computer, by:</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including providing access to code that, after use by the client computer (e.g., device that receives the HTML pages, etc.), results in the client computer operating in accordance with a second protocol (e.g., QUIC, etc.) that is separate from the TCP, in order to establish a second protocol connection (e.g., a QUIC connection, etc.) with another server (e.g., site, etc.) computer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“Google AdSense provides a way for publishers to earn money from their online content. AdSense works by matching ads to your site based on your content and visitors. ... You make your ad spaces available by pasting ad code on your site, and choose where you want the ads to appear.” https://support.google.com/adsense/answer/6242051?hl=en#zippy=%2Cwhat-is-adsense%2Cchow-does-adsense-differ-from-other-ad-networks</p> <p>Note: As set forth below, QUIC is separate from TCP.</p> <p>1. Introduction</p> <p>QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.</p> <p>QUIC implements techniques learned from experience with TCP, SCTP and</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p>
---	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>7.2. Version Negotiation</p> <p>QUIC's connection establishment begins with version negotiation, since all communication between the endpoints, including packet and frame formats, relies on the two endpoints agreeing on a version.</p> <p>A QUIC connection begins with a client sending a Client Initial packet (Section 5.4.1). The details of the handshake mechanisms are described in Section 7.3, but all of the initial packets sent from the client to the server MUST use the long header format - which includes the version of the protocol being used - and they MUST be padded to at least 1200 octets.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. However, when the server generates a Version Negotiation packet, it cannot randomly generate a reserved version number. This is because the server is required to include the same value in its transport parameters (see Section 7.4.4). To avoid the selected version number</p> <ul style="list-style-type: none"> o The initial handshake packet from the client needs to fit in a single packet (#338) <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is "set up" using the aforementioned handshake.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>3.1. Low-Latency Connection Establishment</p> <p>QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
identifying idle information for detecting an idle time period, after which, the second protocol connection is subject to deactivation,	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including identifying idle information for detecting an idle time period, after which, the second protocol connection (e.g., QUIC connection, etc.) is subject to deactivation.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	+-----+-----+-----+		
	Value	Parameter Name	Specification
	0x0000	original_connection_id	Section 18.1
	0x0001	idle_timeout	Section 18.1
	0x0002	stateless_reset_token	Section 18.1
	0x0003	max_packet_size	Section 18.1
	0x0004	initial_max_data	Section 18.1
	0x0005	initial_max_stream_data_bidi_local	Section 18.1
	0x0006	initial_max_stream_data_bidi_remote	Section 18.1
	0x0007	initial_max_stream_data_uni	Section 18.1
	0x0008	initial_max_streams_bidi	Section 18.1
	0x0009	initial_max_streams_uni	Section 18.1
	0x000a	ack_delay_exponent	Section 18.1
	0x000b	max_ack_delay	Section 18.1
	0x000c	disable_migration	Section 18.1
	0x000d	preferred_address	Section 18.1
	0x000e	active_connection_id_limit	Section 18.1
	+-----+-----+-----+		
	Table 6: Initial QUIC Transport Parameters Entries		

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO)."</p> <p><u>"The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried."</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
generating a second protocol packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information, and	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including generating a second protocol packet (e.g., a handshake packet, etc.) including an idle time period parameter field identifying metadata (e.g., a value, etc.) for the idle time period based on the idle information (e.g., the value in the idle_timeout parameter field, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>1. Introduction</p> <p>QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.</p> <p><u>QUIC implements techniques learned from experience with TCP, SCTP and</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>7.2. Version Negotiation</p> <p>QUIC’s connection establishment begins with version negotiation, since all communication between the endpoints, including packet and frame formats, relies on the two endpoints agreeing on a version.</p> <p>A QUIC connection begins with a client sending a Client Initial packet (Section 5.4.1). The details of the handshake mechanisms are described in Section 7.3, but all of the initial packets sent from the client to the server MUST use the long header format - which includes the version of the protocol being used - and they MUST be padded to at least 1200 octets.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. However, when the server generates a Version Negotiation packet, it cannot randomly generate a reserved version number. This is because the server is required to include the same value in its transport parameters (see Section 7.4.4). To avoid the selected version number</p> <ul style="list-style-type: none"> o The initial handshake packet from the client needs to fit in a single packet (#338) <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>3.1. Low-Latency Connection Establishment</p> <p>QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>sending, from the client computer to the another server computer, the second protocol packet to provide the metadata for the idle time period to the another server computer, for use by the another server computer in creating or modifying, based on the metadata, a timeout attribute associated with the second protocol connection.</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a computer-implemented method including sending, from the client computer (e.g., device that receives the HTML pages, etc.) to the another server (e.g., site, etc.) computer, the second protocol packet (e.g., the handshake packet, etc.) to provide the metadata (e.g., the value, etc.) for the idle time period to the another server (e.g., site, etc.) computer, for use by the another server (e.g., site, etc.) computer in creating or modifying, based on the metadata, a timeout attribute associated with the second protocol connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“Google AdSense provides a way for publishers to earn money from their online content. AdSense works by matching ads to your site based on your content and visitors. ... You make your ad spaces available by pasting ad code on your site, and choose where you want the ads to appear.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

	<p>https://support.google.com/adsense/answer/6242051?hl=en#zippy=%2Cwhat-is-adsense%2Cchow-does-adsense-differ-from-other-ad-networks</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter</u> (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 64 Elements	Applicability
<p>The computer-implemented method of claim 34 wherein the access to the code is provided via hypertext.</p>	<p>Google infringes claim 34 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the access to the code is provided via hypertext (e.g., a hypertext transfer protocol (HTTP) link, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“3. Stream States</p> <p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p> <p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2].</u> This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state.”</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 65 Elements	Applicability
<p>The computer-implemented method of claim 64 wherein the hypertext includes a hypertext transfer protocol (HTTP) link.</p>	<p>Google infringes claim 64 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the hypertext includes a hypertext transfer protocol (HTTP) link.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“3. Stream States</p> <p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p> <p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2].</u> This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state.”</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

	https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 66 Elements	Applicability
<p>The computer-implemented method of claim 34 wherein the code is configured to be used by the client computer via hypertext.</p>	<p>Google infringes claim 34 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the code is configured to be used by the client computer via hypertext (e.g., a hypertext transfer protocol (HTTP) link, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“3. Stream States</p> <p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p> <p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2].</u> This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

	mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state." https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 67 Elements	Applicability
<p>The computer-implemented method of claim 66 wherein the hypertext includes a hypertext transfer protocol (HTTP) link.</p>	<p>Google infringes claim 66 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the hypertext includes a hypertext transfer protocol (HTTP) link.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“3. Stream States</p> <p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p> <p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2].</u> This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state.”</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

	https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 68 Elements	Applicability
<p>The computer-implemented method of claim 34 wherein the access to the code is provided via the hypertext transfer protocol (HTTP).</p>	<p>Google infringes claim 34 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the access to the code is provided via the hypertext transfer protocol (HTTP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“3. Stream States</p> <p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p> <p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2].</u> This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state.”</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

	https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,069,945

Claim 69 Elements	Applicability
<p>The computer-implemented method of claim 68 wherein the hypertext transfer protocol (HTTP) transports hypertext that includes a hypertext transfer protocol (HTTP) link.</p>	<p>Google infringes claim 68 and, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform that performs a method wherein the hypertext transfer protocol (HTTP) transports hypertext that includes a hypertext transfer protocol (HTTP) link.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“3. Stream States</p> <p>This section describes streams in terms of their send or receive components. <u>Two state machines are described: one for the streams on which an endpoint transmits data (Section 3.1), and another for streams on which an endpoint receives data (Section 3.2).</u></p> <p>Unidirectional streams use the applicable state machine directly. <u>Bidirectional streams use both state machines.</u> For the most part, the use of these state machines is the same whether the stream is unidirectional or bidirectional. The conditions for opening a stream are slightly more complex for a bidirectional stream because the opening of either send or receive sides causes the stream to open in both directions.</p> <p>...</p> <p><u>3.4. Bidirectional Stream States</u></p> <p>A bidirectional stream is composed of sending and receiving parts. Implementations may represent states of the bidirectional stream as composites of sending and receiving stream states. The simplest model presents the stream as "open" when either sending or receiving parts are in a non-terminal state and "closed" when both sending and receiving streams are in terminal states.</p> <p>Table 2 shows a more complex mapping of <u>bidirectional stream states that loosely correspond to the stream states in HTTP/2 [HTTP2].</u> This shows that multiple states on sending or receiving parts of streams are mapped to the same composite state. Note that this is just one possibility for such a mapping; this mapping requires that data is acknowledged before the transition to a "closed" or "half-closed" state.”</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,069,945

	https://tools.ietf.org/html/draft-ietf-quic-transport-22 (emphasis added)
--	--

Caveat: The notes and/or cited excerpts utilized herein are set forth for illustrative purposes only and are not meant to be limiting in any manner. For example, the notes and/or cited excerpts, may or may not be supplemented or substituted with different excerpt(s) of the relevant reference(s), as appropriate. Further, to the extent any error(s) and/or omission(s) exist herein, all rights are reserved to correct the same.

JENAM TECH, LLC'S CORRECTED FIRST AMENDED SET OF INFRINGEMENT CONTENTIONS

U.S. Patent No. 9,923,995 – Google LLC Claims 1, 11, 12, 13, 16, and 21

Jenam Tech LLC (“Jenam”) provides evidence of infringement of claims 1, 11, 12, 13, 16, and 21 of U.S. Patent No. 9,923,995 (hereinafter “the ’995 patent”) by Google LLC (“Google”). In support thereof, Jenam provides the following claim charts.

“Accused Instrumentalities” as used herein refers to at least one or more websites or web addresses including, but not limited to www.google.com, stored and/or hosted on one or more servers owned or under the control of Google. For the sake of clarity, Jenam alleges that all of Google’s various websites, products, and platforms utilizing QUIC infringe the ’995 patent, including, but not limited to Google Edge Network, Google Cloud, Chrome Enterprise, G suite, Google Play, Chrome, Android (Android Enterprise, Android Messages (RCS)), Duo, Google Ads, Adwords, Google Analytics, Youtube, Google Mobile apps, Google shopping, and Google Maps. A list of Google’s “products” can be found at https://about.google/intl/en_us/products/. On information and belief, the Accused Instrumentalities referred to in the charts below are representative of Google’s use of QUIC in Google’s other websites, products and platforms. These claim charts demonstrate Google’s infringement, and provide notice of such infringement, by comparing each element of the asserted claims to corresponding components, aspects, and/or features of the Accused Instrumentalities. These claim charts are not intended to constitute an expert report on infringement. These claim charts include information provided by way of example, and not by way of limitation.

The analysis set forth below is based only upon information from publicly available resources regarding the Infringing Instrumentalities. An analysis of Google’s (or other third parties’) technical documentation and/or software source code may assist in fully identify all infringing features and functionality. Accordingly, Jenam reserves the right to supplement this infringement analysis once such information is made available to Jenam. Furthermore, Jenam reserves the right to revise this infringement analysis, as appropriate, upon issuance of a court order construing any terms recited in the asserted claims.

Unless otherwise noted, Jenam contends that Google directly infringes the ’995 patent in violation of 35 U.S.C. § 271(a) by selling, offering to sell, making, using, and/or importing the Infringing Instrumentalities. The following exemplary analysis demonstrates that infringement. Unless otherwise noted, Jenam further contends that the evidence below supports a finding of indirect infringement under 35 U.S.C. §§ 271(b) and/or (c), in conjunction with other evidence of liability under one or more of those subsections. Google makes, uses, sells, imports, or offers for sale in the United States, or has made, used, sold, imported, or offered for sale in the past, without authority, or induces others to make, use, sell, import, or offer for sale in the United States, or has induced others to make, use, sell, import, or offer for sale in the past, without authority products, equipment, or services that infringe claims 1, 11, 12, 13, 16, and 21 of the ’995 patent, including without limitation, the Accused Instrumentalities.

Unless otherwise noted, Jenam believes and contends that each element of each claim asserted herein is literally met through Google’s provision of the Infringing Instrumentalities. However, to the extent that Google attempts to allege that any asserted claim element is not literally met, Jenam believes and contends that such elements are met under the doctrine of equivalents. More

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

specifically, in its investigation and analysis of the Infringing Instrumentalities, Jenam did not identify any substantial differences between the elements of the patent claims and the corresponding features of the Infringing Instrumentalities, as set forth herein. In each instance, the identified feature of the Infringing Instrumentalities performs at least substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim element.

To the extent the chart of an asserted claim relies on evidence about certain specifically-identified Accused Instrumentalities, Jenam asserts that, on information and belief, any similarly-functioning instrumentalities also infringes the charted claim. Jenam reserves the right to amend this infringement analysis based on other products made, used, sold, imported, or offered for sale by Google. Jenam also reserves the right to amend this infringement analysis by citing other claims of the '995 patent, not listed in the claim chart, that are infringed by the Accused Instrumentalities. Jenam further reserves the right to amend this infringement analysis by adding, subtracting, or otherwise modifying content in the "Accused Instrumentalities" column of each chart.

Claim 1 Elements	Applicability
<p>An apparatus comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for:</p> <p>receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established;</p>	<p>Google uses an apparatus (e.g., one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) including a non-transitory memory storing instructions, and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for communicating using the QUIC protocol which, in turn, mandates receipt, by a second node (e.g., one of a QUIC-compliant server or client, etc.) from a first node (e.g., the other one of the QUIC-compliant server or client, etc.), a transmission control protocol (TCP)-variant packet (e.g., QUIC negotiation packet, etc.) in advance of a TCP-variant connection (e.g., QUIC connection, etc.) being established.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their "Google Cloud Platform" uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>[W]e're happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:

“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube”

https://en.wikipedia.org/wiki/Google_Cloud_Platform

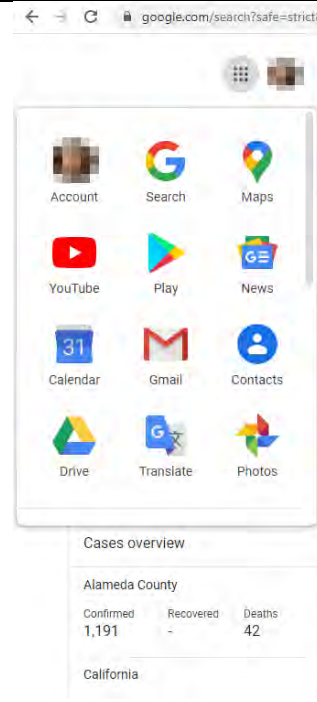
Note: At least a portion of the citations herein are made to the QUIC standard found at: <https://tools.ietf.org/html/draft-ietf-quic-transport-22>, and/or <https://tools.ietf.org/html/draft-ietf-quic-transport-27>. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:

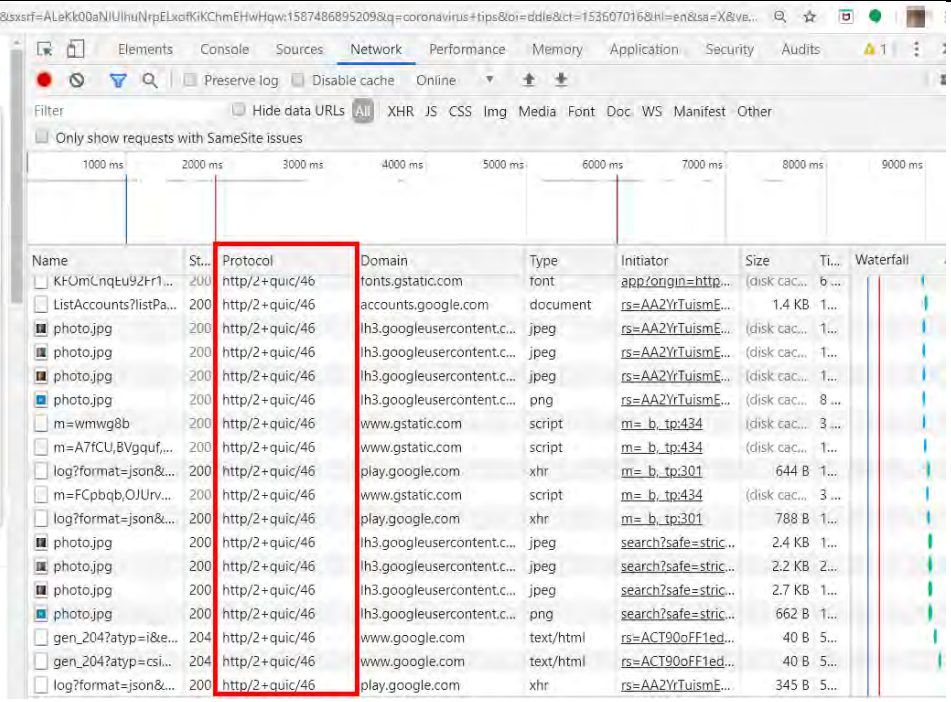
<https://tools.ietf.org/html/draft-ietf-quic-transport-00>
<https://tools.ietf.org/html/draft-ietf-quic-transport-01>
<https://tools.ietf.org/html/draft-ietf-quic-transport-02>
<https://tools.ietf.org/html/draft-ietf-quic-transport-03>
<https://tools.ietf.org/html/draft-ietf-quic-transport-04>
<https://tools.ietf.org/html/draft-ietf-quic-transport-05>
<https://tools.ietf.org/html/draft-ietf-quic-transport-06>
<https://tools.ietf.org/html/draft-ietf-quic-transport-07>
<https://tools.ietf.org/html/draft-ietf-quic-transport-08>
<https://tools.ietf.org/html/draft-ietf-quic-transport-09>
<https://tools.ietf.org/html/draft-ietf-quic-transport-10>
<https://tools.ietf.org/html/draft-ietf-quic-transport-11>
<https://tools.ietf.org/html/draft-ietf-quic-transport-12>
<https://tools.ietf.org/html/draft-ietf-quic-transport-13>
<https://tools.ietf.org/html/draft-ietf-quic-transport-14>
<https://tools.ietf.org/html/draft-ietf-quic-transport-15>
<https://tools.ietf.org/html/draft-ietf-quic-transport-16>
<https://tools.ietf.org/html/draft-ietf-quic-transport-17>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> <u>Note:</u> Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities. </p> <p> <u>Note:</u> Below is a web page of Google (https://www.google.com/). </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995





Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfUmCnqtu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app:origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
m=A7fCU,BVgqf...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, QUIC is a “variant” of TCP.

1. Introduction

QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.

QUIC implements techniques learned from experience with TCP, SCTP and QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>7.2. Version Negotiation</p> <p>QUIC’s connection establishment begins with version negotiation, since all communication between the endpoints, including packet and frame formats, relies on the two endpoints agreeing on a version.</p> <p>A QUIC connection begins with a client sending a Client Initial packet (Section 5.4.1). The details of the handshake mechanisms are described in Section 7.3, but all of the initial packets sent from the client to the server MUST use the long header format - which includes the version of the protocol being used - and they MUST be padded to at least 1200 octets.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. However, when the server generates a Version Negotiation packet, it cannot randomly generate a reserved version number. This is because the server is required to include the same value in its transport parameters (see Section 7.4.4). To avoid the selected version number</p> <ul style="list-style-type: none"> o The initial handshake packet from the client needs to fit in a single packet (#338) <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>3.1. Low-Latency Connection Establishment</p> <p>QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>detecting an idle time period parameter field in the TCP-variant packet;</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) to detect an idle time period parameter field (e.g., idle timeout parameter field, etc.) in the TCP-variant packet (e.g., QUIC negotiation packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>The format of the transport parameters is the TransportParameters struct from Figure 6. This is described using the presentation language from Section 3 of [I-D.ietf-tls-tls13].</p> <pre> uint32 QuicVersion; enum { initial_max_stream_data(0), initial_max_data(1), initial_max_stream_id(2), idle_timeout(3), omit_connection_id(4), max_packet_size(5), stateless_reset_token(6), (65535) } TransportParameterId; </pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>identifying metadata in the idle time period parameter field for an idle time period that is detectable by the first node and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active; and</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) to identify metadata (e.g., a value, etc.) in the idle time period parameter field (e.g., idle timeout parameter field, etc.) for an idle time period that is detectable by the first node (e.g., the other one of the QUIC-compliant server or client, etc.) and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <pre> idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes). </pre>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>modifying, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection.</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) to modify, by the second node (e.g., one of the QUIC-compliant server or client, etc.) and based on the metadata (e.g., the value of the idle timeout parameter field, etc.), a timeout attribute associated with the TCP-variant connection.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,995

	<p>7.8. Connection Termination</p> <p>Connections should remain open until they become idle for a pre-negotiated period of time. A QUIC connection, once established, can be terminated in one of three ways:</p> <p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

Claim 11 Elements	Applicability
<p>The apparatus of claim 1 wherein the one or more processors execute the instructions for:</p> <p>detecting the idle time period based on the timeout attribute; and</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) to detect an idle time period based on the timeout attribute.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-27</p>
<p>in response to detecting the idle time period, deactivating the TCP-variant connection by releasing a resource allocated for the TCP-variant connection by one of the first and second nodes without signaling another one of the first and second nodes that is related to the detection of the idle time period.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) to detect the idle time period and, in response thereto, deactivate (e.g., terminate the connection, etc.) the TCP-variant connection by releasing a resource allocated for the TCP-variant connection (e.g., the QUIC connection, etc.) by one of the first and second nodes without signaling another one of the first and second nodes that is related to the detection of the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

Claim 12 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured such that at least one of the detecting or the identifying is performed at a TCP-variant layer other than a TCP layer, where the TCP-variant layer is above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) configured such that at least one of the detecting or the identifying is performed at a TCP-variant layer (e.g., QUIC protocol layer, etc.) other than a TCP layer, where the TCP-variant layer is above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC is a “variant” of TCP.</p> <p>“1. Introduction</p> <p>QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].</p> <p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.”</p> <p>https://medium.com/@nirosh/understanding-quic-wire-protocol-d0ff97644de7</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

Claim 13 Elements	Applicability
<p>The apparatus of claim 1 wherein the one or more processors execute a network application that is configured to perform a 3-way TCP handshake for establishing a TCP connection that is different than the TCP-variant connection, and wherein the network application is configured to establish the TCP-variant connection instead of the TCP connection in order to permit negotiation, between the first node and the second node, of the timeout attribute, where the timeout attribute is not available when establishing the TCP connection, but is available when establishing the TCP-variant connection so that the TCP-variant connection is capable of being at least partially closed when inactive based on the timeout attribute.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., the one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) to perform a 3-way TCP handshake for establishing a TCP connection that is different than the TCP-variant connection (e.g., the QUIC connection, etc.), and wherein the network application is configured to establish the TCP-variant connection instead of the TCP connection in order to permit negotiation, between the first node (e.g., the other one of the QUIC-compliant server or client, etc.) and the second node (e.g., one of the QUIC-compliant server or client, etc.), of the timeout attribute, where the timeout attribute is not available when establishing the TCP connection, but is available when establishing the TCP-variant connection so that the TCP-variant connection is capable of being at least partially closed (e.g., the connection is terminated, etc.) when inactive based on the timeout attribute.</p> <p>Note: In addition to the QUIC protocol, the apparatus also operates in accordance with the standard transmission control protocol (TCP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side’s initial sequence number and send a confirming acknowledgment.</p> <p>1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y</p> <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3]."</p> <p>"Request for Comments" (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter that is detected by a recipient of such packet.</p> <p>"7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,995

	<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
--	---

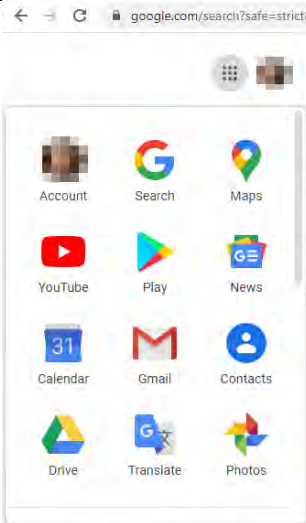
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

Claim 15 Elements	Applicability
<p>An apparatus comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for:</p> <p>receiving idle information for detecting an idle time period, during which, no packet is communicated in a transmission control protocol (TCP)-variant connection to keep the TCP-variant connection active;</p>	<p>Google uses an apparatus (e.g., one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) including a non-transitory memory storing instructions, and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for communicating using the QUIC protocol which, in turn, mandates receipt of idle information for detecting an idle time period, during which, no packet is communicated in the TCP-variant connection (e.g., the QUIC connection, etc.) to keep the TCP-variant connection active.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>[W]e’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p>Note: Below is a web page of Google (https://www.google.com/).</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

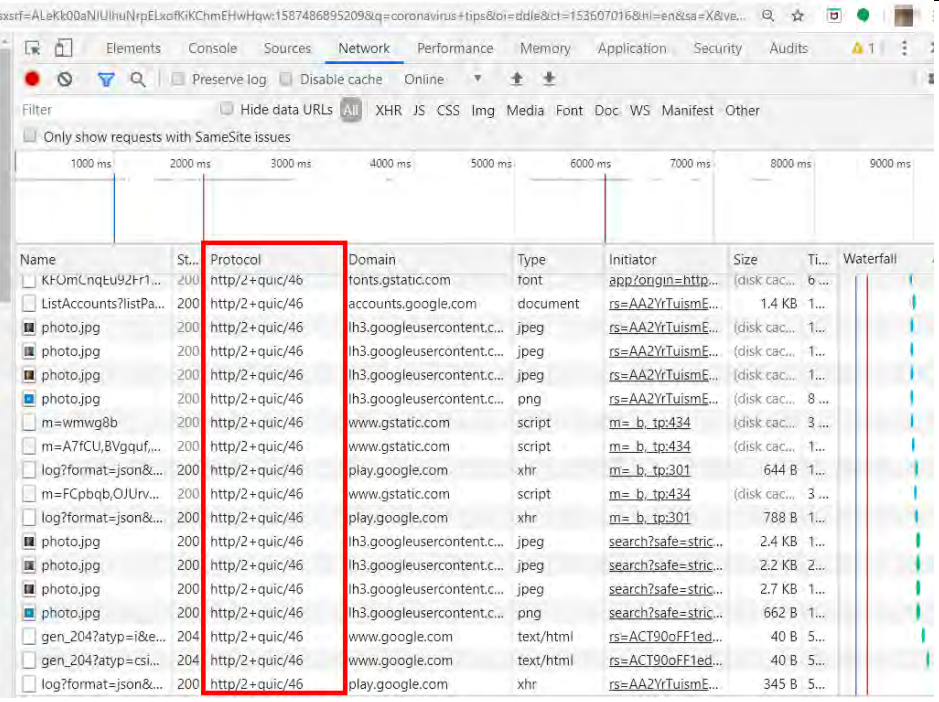


Cases overview

Alameda County

Confirmed	Recovered	Deaths
1,191	-	42

California



Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfUmCnqeu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app:origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
m=A7fCU,BVgqf...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	644 B	1...	
m=FCpbqb,OJUrv...	200	http/2+quic/46	www.gstatic.com	script	m= b_ tp:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ tp:301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, a QUIC is a “variant” of TCP.

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

“1. Introduction

QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].

QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
generating a TCP-variant packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information; and	<p>Google uses the apparatus (e.g., the one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) to generate a TCP-variant packet (e.g., QUIC negotiation packet, etc.) including an idle time period parameter field (e.g., idle_timeout parameter field, etc.) identifying metadata (e.g., a value, etc.) for the idle time period based on the idle information.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>sending, from a first node to a second node, the TCP-variant packet in advance of the TCP-variant connection being established to provide the metadata for the idle time period to the second node, for use by the second node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) to send, from a first node (e.g., the other one of the QUIC-compliant server or client, etc.) to a second node (e.g., one of a QUIC-compliant server or client, etc.), a transmission control protocol (TCP)-variant packet (e.g., QUIC negotiation packet, etc.) in advance of a TCP-variant connection (e.g., QUIC connection, etc.) being established to provide the metadata (e.g., the value, etc.) for the idle time period to the second node, for use by the second node in modifying, based on the metadata, a timeout attribute associated with the TCP-variant connection.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>“5.2.2. Server Packet Handling</p> <p>...</p> <p>If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected.</p> <p>...</p> <p>If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets.</p> <p>...</p> <p>7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,995

	comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,995

	<p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the <code>idle_timeout</code> transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

Claim 16 Elements	Applicability
<p>The apparatus of claim 15 wherein the apparatus is configured such that modifying the timeout attribute reduces a number of keep-alive signals that are required to be communicated.</p>	<p>Google infringes claim 15 and uses the apparatus (e.g., the one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) to reduce a number of keep-alive signals (e.g., keep-alive packets, etc.) that are required to be communicated.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, keep-alive signals are not required as a result of using idle timers and reducing the number of keep-alive signals to zero and, without a negotiated idle timeout, neither endpoint of a connection knows when the other side might timeout, so keep-alives (e.g. PINGs) have to be sent at unnecessarily shorter intervals to insure the connection to stay active. Further, with a negotiated idle time period, only one side sends one or more keep-alives, if a keep-alive is used.</p> <p>The QUIC IETF transport spec recognizes the need to avoid unnecessary keep-alives (e.g. PINGs). See section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2 and section 5.1 of the IETF HTTP/3 draft at https://tools.ietf.org/id/draft-ietf-quic-http-34.html#name-idle-connections.</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.” https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p> <p>“5.1. Idle Connections</p> <p>Each QUIC endpoint declares an idle timeout during the handshake. If the connection remains idle (no packets received) for longer than this duration, the peer will assume that the connection has been closed. HTTP/3 implementations will need to open a new connection for new requests if the existing connection has been idle for longer than the server's advertised idle timeout, and SHOULD do so if approaching the idle timeout.</p> <p>HTTP clients are expected to request that the transport keep connections open while there are responses outstanding for requests or server pushes, as described in Section 19.2 of [QUIC-TRANSPORT]. If the client is not expecting a response from the server, allowing an idle connection to time out is preferred over expending effort maintaining a connection that might not be needed. A gateway MAY maintain connections in anticipation of need rather than incur the latency cost of connection establishment to servers. Servers SHOULD NOT actively keep connections open.” https://tools.ietf.org/id/draft-ietf-quic-http-27.html</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2)
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<ul style="list-style-type: none"> • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2)."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>"10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO)."</p> <p>"The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>"3.3. Session resumption versus Keep-alive</p> <p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods. • Resume the session after a long idle period, using 0-RTT resumption when appropriate. <p>The first strategy is the easiest, but it only applies to certain applications. Either the server or the client in a QUIC application can send PING frames as keep-alives, to prevent the connection and any on-path state from timing out.</p> <p><u>Recommendations for the use of keep-alives are application specific, mainly depending on the latency requirements and message frequency of the application. In this case, the application mapping must specify whether the client or server is responsible for keeping the application alive.</u> Note that sending PING frames more frequently than every 30 seconds over long idle periods may result in a too much unproductive traffic and power usage for some situations.</p> <p><u>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic.</u> In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with restarting the connection. This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.</p> <p>The tradeoffs between resumption and keepalive need to be evaluated on a per-application basis. However, in general applications should use keepalives only in circumstances where continued communication is highly likely; [QUIC-HTTP], for instance, recommends using PING frames for keepalive only when a request is outstanding.” https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

Claim 21 Elements	Applicability
<p>The apparatus of claim 15 wherein the one or more processors execute the instructions for: detecting the idle time period based on the idle information; and in response to detecting the idle time period, deactivating the TCP-variant connection.</p>	<p>Google infringes claim 15 and uses the apparatus (e.g., the one or more servers and/or Google Cloud Platform, or at least one component thereof, etc.) to detect the idle time period based on the idle information and, in response thereto, deactivate (e.g., terminate the connection, etc.) the TCP-variant connection (e.g., QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,995

	<p>initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p> <p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	---

Caveat: The notes and/or cited excerpts utilized herein are set forth for illustrative purposes only and are not meant to be limiting in any manner. For example, the notes and/or cited excerpts, may or may not be supplemented or substituted with different excerpt(s) of the relevant reference(s), as appropriate. Further, to the extent any error(s) and/or omission(s) exist herein, all rights are reserved to correct the same.

JENAM TECH, LLC'S FIRST AMENDED SET OF INFRINGEMENT CONTENTIONS

**U.S. Patent No. 9,923,996 – Google LLC
Claims 2, 12, 22, 23, 27, and 30**

Jenam Tech LLC (“Jenam”) provides evidence of infringement of claims 2, 12, 22, 23, 27, and 30 of U.S. Patent No. 9,923,996 (hereinafter “the ’996 patent”) by Google LLC (“Google”). In support thereof, Jenam provides the following claim charts.

“Accused Instrumentalities” as used herein refers to at least one or more websites or web addresses including, but not limited to www.google.com, stored and/or hosted on one or more servers owned or under the control of Google. For the sake of clarity, Jenam alleges that all of Google’s various websites, products, and platforms utilizing QUIC infringe the ’996 patent, including, but not limited to Google Edge Network, Google Cloud, Chrome Enterprise, G suite, Google Play, Chrome, Android (Android Enterprise, Android Messages (RCS)), Duo, Google Ads, Adwords, Google Analytics, Youtube, Google Mobile apps, Google shopping, and Google Maps. A list of Google’s “products” can be found at https://about.google/intl/en_us/products/. On information and belief, the Accused Instrumentalities referred to in the charts below are representative of Google’s use of QUIC in Google’s other websites, products and platforms. These claim charts demonstrate Google’s infringement, and provide notice of such infringement, by comparing each element of the asserted claims to corresponding components, aspects, and/or features of the Accused Instrumentalities. These claim charts are not intended to constitute an expert report on infringement. These claim charts include information provided by way of example, and not by way of limitation.

The analysis set forth below is based only upon information from publicly available resources regarding the Infringing Instrumentalities. An analysis of Google’s (or other third parties’) technical documentation and/or software source code may assist in fully identify all infringing features and functionality. Accordingly, Jenam reserves the right to supplement this infringement analysis once such information is made available to Jenam. Furthermore, Jenam reserves the right to revise this infringement analysis, as appropriate, upon issuance of a court order construing any terms recited in the asserted claims.

Unless otherwise noted, Jenam contends that Google directly infringes the ’996 patent in violation of 35 U.S.C. § 271(a) by selling, offering to sell, making, using, and/or importing the Infringing Instrumentalities. The following exemplary analysis demonstrates that infringement. Unless otherwise noted, Jenam further contends that the evidence below supports a finding of indirect infringement under 35 U.S.C. §§ 271(b) and/or (c), in conjunction with other evidence of liability under one or more of those subsections. Google makes, uses, sells, imports, or offers for sale in the United States, or has made, used, sold, imported, or offered for sale in the past, without authority, or induces others to make, use, sell, import, or offer for sale in the United States, or has induced others to make, use, sell, import, or offer for sale in the past, without authority products, equipment, or services that infringe claims 2, 12, 22, 23, 27, and 30 of the ’996 patent, including without limitation, the Accused Instrumentalities.

Unless otherwise noted, Jenam believes and contends that each element of each claim asserted herein is literally met through Google’s provision of the Infringing Instrumentalities. However, to the extent that Google attempts to allege that any asserted claim element is not literally met, Jenam believes and contends that such elements are met under the doctrine of equivalents. More

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

specifically, in its investigation and analysis of the Infringing Instrumentalities, Jenam did not identify any substantial differences between the elements of the patent claims and the corresponding features of the Infringing Instrumentalities, as set forth herein. In each instance, the identified feature of the Infringing Instrumentalities performs at least substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim element.

To the extent the chart of an asserted claim relies on evidence about certain specifically-identified Accused Instrumentalities, Jenam asserts that, on information and belief, any similarly-functioning instrumentalities also infringes the charted claim. Jenam reserves the right to amend this infringement analysis based on other products made, used, sold, imported, or offered for sale by Google. Jenam also reserves the right to amend this infringement analysis by citing other claims of the '996 patent, not listed in the claim chart, that are infringed by the Accused Instrumentalities. Jenam further reserves the right to amend this infringement analysis by adding, subtracting, or otherwise modifying content in the "Accused Instrumentalities" column of each chart.

Claim 1 Elements	Applicability
An apparatus comprising: a non-transitory memory storing a network application; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application such that the network application is configured to operate in accordance with a first protocol including a transmission control protocol (TCP), the apparatus, when operating in accordance with the first protocol to establish a TCP connection, configured to:	<p>Google uses an apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) including a non-transitory memory storing a network application (e.g., server software, etc.), and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application such that the network application is configured to operate in accordance with a first protocol including a transmission control protocol (TCP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their "Google Cloud Platform" uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>[W]e're happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The "Google Cloud Platform" that, as established above, uses QUIC, is also used for ALL of Google's services:</p>

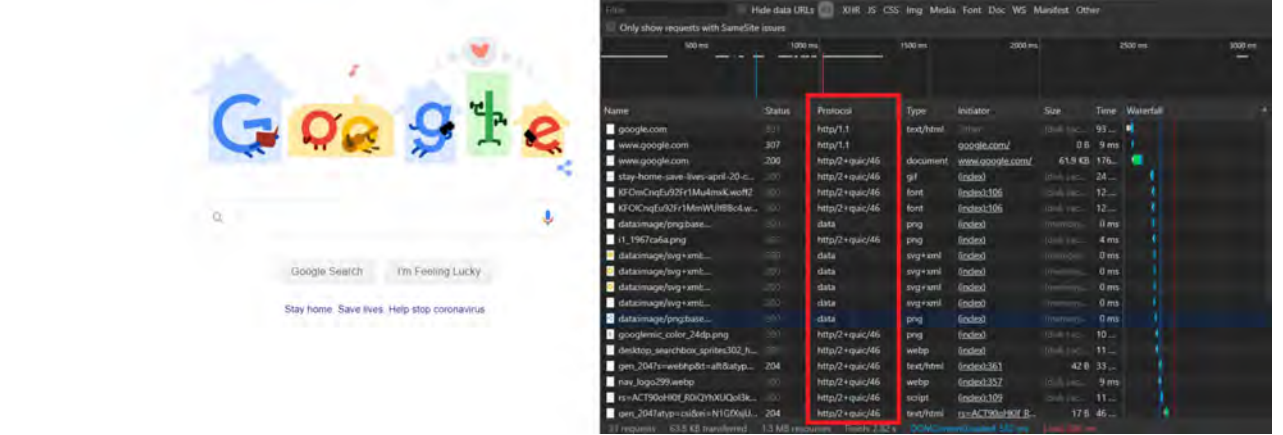
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p> https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 </p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities. </p> <p> Note: Below is a web page of Google (https://www.google.com/). </p>
--	---

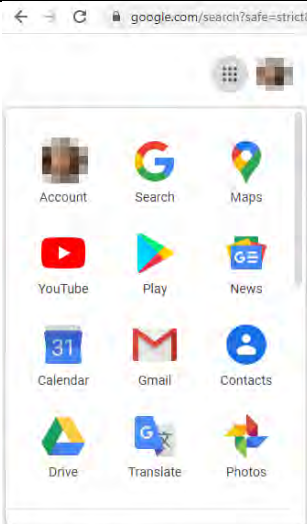
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	 <p>The image shows the Google homepage on the left and a network waterfall chart on the right. The waterfall chart displays various requests to google.com, including HTML documents, images, and scripts. A red box highlights the first few requests, which are the initial synchronization and acknowledgment steps of a QUIC connection.</p>
<p>communicate a segment including at least one first synchronize bit;</p> <p>communicate a first acknowledgement of the segment, and at least one second synchronize bit; and</p> <p>communicate a second acknowledgement;</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) that, when operating in accordance with the first protocol to establish the TCP connection, is configured for a) communicating a segment including at least one first synchronize bit; b) communicating a first acknowledgement of the segment, and at least one second synchronize bit; and c) communicating a second acknowledgement.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a) is met by 1); b) is met by 2)/3), and c) is met by 4).</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <ol style="list-style-type: none"> 1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3]."</p> <p>"Request for Comments" (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
<p>wherein the network application is further configured to operate in accordance with a second protocol that is separate from the TCP, the apparatus, when operating in accordance with the second protocol to establish a second protocol connection, configured to:</p> <p>receive, by a second node from a first node, a packet;</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the network application is further configured to operate in accordance with a second protocol (e.g., QUIC, etc.) that is separate from the TCP, the apparatus, when operating in accordance with the second protocol to establish a second protocol connection (e.g., QUIC connection, etc.), configured to: receive, by a second node (e.g., one of a QUIC-compliant server or client, etc.) from a first node (e.g., the other one of the QUIC-compliant server or client, etc.), a packet (e.g., QUIC negotiation packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

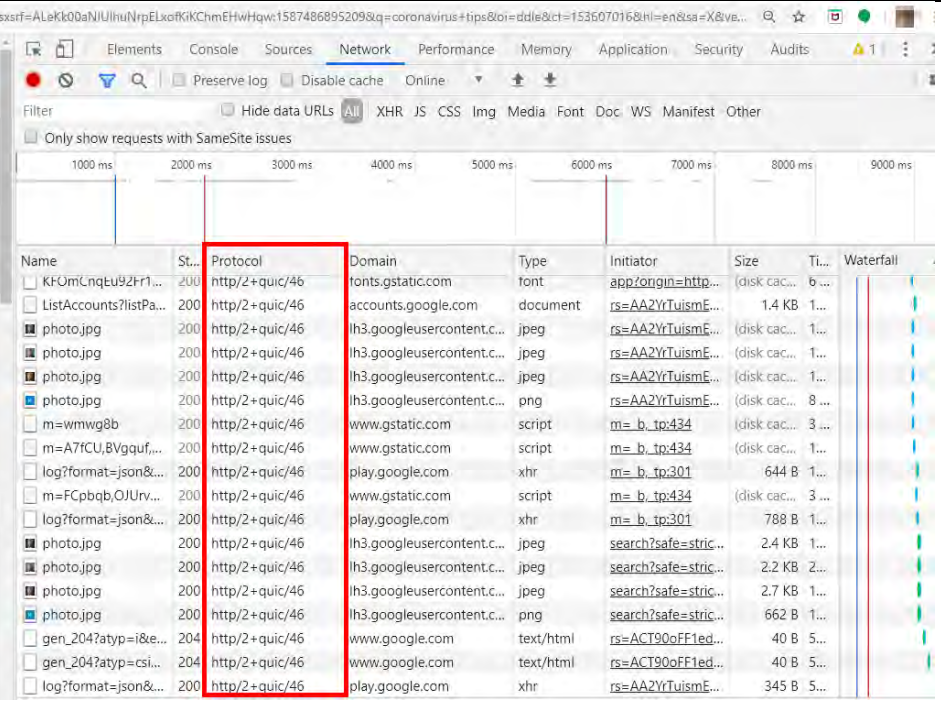


Cases overview

Alameda County

Confirmed	Recovered	Deaths
1,191	-	42

California



Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfOmCnqEu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app:origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m= b_ip:434	(disk cac...	3...	
m=A7fCU,BVgquf,...	200	http/2+quic/46	www.gstatic.com	script	m= b_ip:434	(disk cac...	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m= b_ip:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ip:301	644 B	1...	
log?format=json&...	200	http/2+quic/46	www.gstatic.com	script	m= b_ip:434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m= b_ip:301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stic...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stic...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stic...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stic...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, QUIC is separate from TCP.

1. Introduction

QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.

QUIC implements techniques learned from experience with TCP, SCTP and

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-07>

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>7.2. Version Negotiation</p> <p>QUIC’s connection establishment begins with version negotiation, since all communication between the endpoints, including packet and frame formats, relies on the two endpoints agreeing on a version.</p> <p>A QUIC connection begins with a client sending a Client Initial packet (Section 5.4.1). The details of the handshake mechanisms are described in Section 7.3, but all of the initial packets sent from the client to the server MUST use the long header format - which includes the version of the protocol being used - and they MUST be padded to at least 1200 octets.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. However, when the server generates a Version Negotiation packet, it cannot randomly generate a reserved version number. This is because the server is required to include the same value in its transport parameters (see Section 7.4.4). To avoid the selected version number</p> <ul style="list-style-type: none"> o The initial handshake packet from the client needs to fit in a single packet (#338) <p>QUIC packet: A well-formed UDP payload that can be parsed by a QUIC receiver. QUIC packet size in this document refers to the UDP</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>3.1. Low-Latency Connection Establishment</p> <p>QUIC relies on a combined cryptographic and transport handshake for setting up a secure transport connection. QUIC connections are expected to commonly use 0-RTT handshakes, meaning that for most QUIC connections, data can be sent immediately following the client handshake packet, without waiting for a reply from the server. QUIC provides a dedicated stream (Stream ID 0) to be used for performing the cryptographic handshake and QUIC options negotiation. The format of the QUIC options and parameters used during negotiation are described in this document, but the handshake protocol that runs on Stream ID 0 is described in the accompanying cryptographic handshake draft [QUIC-TLS].</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>detect an idle time period parameter field in the packet;</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) that, when operating in accordance with the second protocol (e.g., QUIC, etc.) to establish the second protocol connection (e.g., QUIC connection, etc.), is configured to: detect an idle time period parameter field (e.g., idle timeout parameter field, etc.) in the packet (e.g., QUIC negotiation packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>7.4. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Value	Parameter Name	Specification
0x0000	initial_max_stream_data	Section 7.4.1
0x0001	initial_max_data	Section 7.4.1
0x0002	initial_max_stream_id	Section 7.4.1
0x0003	idle_timeout	Section 7.4.1
0x0004	omit_connection_id	Section 7.4.1
0x0005	max_packet_size	Section 7.4.1
0x0006	stateless_reset_token	Section 7.4.1

Table 4: Initial QUIC Transport Parameters Entries

The format of the transport parameters is the TransportParameters struct from Figure 6. This is described using the presentation language from Section 3 of [\[I-D.ietf-tls-tls13\]](#).

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<pre>uint32 QuicVersion; enum { initial_max_stream_data(0), initial_max_data(1), initial_max_stream_id(2), idle_timeout(3), omit_connection_id(4), max_packet_size(5), stateless_reset_token(6), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
<p>identify metadata in the idle time period parameter field for an idle time period, detectable by the first node, where, after the idle time period is detected, the second protocol connection is deemed inactive; and</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) that, when operating in accordance with the second protocol (e.g., QUIC, etc.) to establish the second protocol connection (e.g., QUIC connection, etc.), is configured to: identify metadata (e.g., a value, etc.) in the idle time period parameter field (e.g., idle timeout parameter field, etc.) for an idle time period, detectable by the first node, where, after the idle time period is detected, the second protocol connection is deemed inactive.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p>
create or modify, by the second node and based on the metadata, a timeout attribute associated with the second protocol connection.	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) that, when operating in accordance with the second protocol (e.g., QUIC, etc.) to establish the second protocol connection (e.g., QUIC connection, etc.), is configured to: create or modify, by the second node and based on the metadata (e.g., the value of the idle timeout parameter field, etc.), a timeout attribute associated with the second protocol connection.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>idle_timeout (0x0003): The idle timeout is a value in seconds that is encoded as an unsigned 16-bit integer. The maximum value is 600 seconds (10 minutes).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>7.8. Connection Termination</p> <p>Connections should remain open until they become idle for a pre-negotiated period of time. A QUIC connection, once established, can be terminated in one of three ways:</p> <p>7.8.2. Idle Timeout</p> <p>A connection that remains idle for longer than the idle timeout (see Section 7.4.1) becomes closed. Either peer removes connection state if they have neither sent nor received a packet for this time.</p> <p>The time at which an idle timeout takes effect won't be perfectly synchronized on peers. A connection enters the draining period when the idle timeout expires. During this time, an endpoint that receives new packets MAY choose to restore the connection. Alternatively, an endpoint that receives packets MAY signal the timeout using an immediate close.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-07</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Claim 2 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured such that creating or modifying the timeout attribute renders one or more keep-alive packets in the second protocol connection unnecessary.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to create or modify the timeout attribute which renders one or more keep-alive packets in the second protocol connection (e.g., the QUIC connection, etc.) unnecessary.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, keep-alive signals are not required as a result of using idle timers and reducing the number of keep-alive signals to zero and, without a negotiated idle timeout, neither endpoint of a connection knows when the other side might timeout, so keep-alives (e.g. PINGs) have to be sent at unnecessarily shorter intervals to insure the connection to stay active. Further, with a negotiated idle time period, only one side sends one or more keep-alives, if a keep-alive is used.</p> <p>The QUIC IETF transport spec recognizes the need to avoid unnecessary keep-alives (e.g. PINGs). See section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2 and section 5.1 of the IETF HTTP/3 draft at https://tools.ietf.org/id/draft-ietf-quic-http-34.html#name-idle-connections.</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows. ” https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p> <p>“5.1. Idle Connections</p> <p>Each QUIC endpoint declares an idle timeout during the handshake. If the connection remains idle (no packets received) for longer than this duration, the peer will assume that the connection has been closed. HTTP/3 implementations will need to open a new connection for new requests if the existing connection has been idle for longer than the server's advertised idle timeout, and SHOULD do so if approaching the idle timeout.</p> <p>HTTP clients are expected to request that the transport keep connections open while there are responses outstanding for requests or server pushes, as described in Section 19.2 of [QUIC-TRANSPORT]. If the client is not expecting a response from the server, allowing an idle connection to time out is preferred over expending effort maintaining a connection that might not be needed. A gateway MAY maintain connections in anticipation of need rather than incur the latency cost of connection establishment to servers. Servers SHOULD NOT actively keep connections open.” https://tools.ietf.org/id/draft-ietf-quic-http-27.html</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily created or modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p style="padding-left: 40px;"><u>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer.</u> The PING frame contains no additional fields.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“3.3. Session resumption versus Keep-alive</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p> <ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods. • Resume the session after a long idle period, using 0-RTT resumption when appropriate. <p>The first strategy is the easiest, but it only applies to certain applications. Either the server or the client in a QUIC application can send PING frames as keep-alives, to prevent the connection and any on-path state from timing out.</p> <p><u>Recommendations for the use of keep-alives are application specific, mainly depending on the latency requirements and message frequency of the application. In this case, the application mapping must specify whether the client or server is responsible for keeping the application alive.</u> Note that sending PING frames more frequently than every 30 seconds over long idle periods may result in a too much unproductive traffic and power usage for some situations.</p> <p><u>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic.</u> In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with restarting the connection. This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.</p> <p>The tradeoffs between resumption and keepalive need to be evaluated on a per-application basis. However, in general applications should use keepalives only in circumstances where continued</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,996

	communication is highly likely; [QUIC-HTTP], for instance, recommends using PING frames for keepalive only when a request is outstanding.” https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Claim 12 Elements	Applicability
<p>The apparatus of claim 1 wherein the apparatus is configured to utilize the second protocol instead of the TCP in order to permit negotiation, between the first node and the second node, of the timeout attribute, where a negotiated timeout attribute is not available when operating in accordance with the TCP, but is available when establishing the second protocol connection so as to permit the second protocol connection to be at least partially closed based on the negotiated timeout attribute when the second protocol connection is inactive.</p>	<p>Google infringes claim 1 and uses the apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to utilize the second protocol (e.g., QUIC, etc.) instead of the TCP in order to permit negotiation, between the first node (e.g., the other one of the QUIC-compliant server or client, etc.) and the second node (e.g., one of the QUIC-compliant server or client, etc.), of the timeout attribute, where a negotiated timeout attribute is not available when operating in accordance with the TCP, but is available when establishing the second protocol connection (e.g., the QUIC connection, etc.) so as to permit the second protocol connection to be at least partially closed (e.g., the connection is terminated, etc.) based on the negotiated timeout attribute when the second protocol connection is inactive.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an <code>idle_timeout</code> parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries
QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,996

	<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
--	---

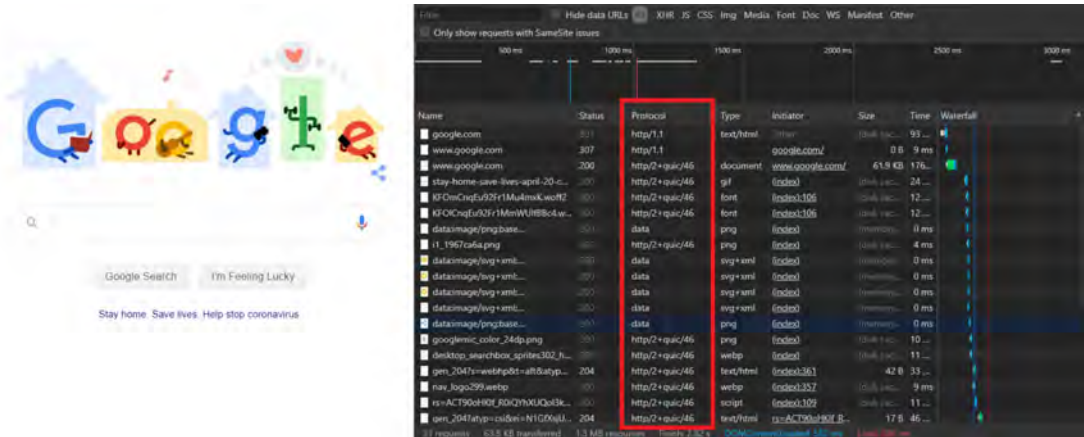
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Claim 15 Elements	Applicability
<p>An apparatus comprising: a non-transitory memory storing a network application; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application such that the network application is configured to operate in accordance with a first protocol including a transmission control protocol (TCP), the apparatus, when operating in accordance with the first protocol to establish a TCP connection, configured to:</p>	<p>Google uses an apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) including a non-transitory memory storing a network application (e.g., server software, etc.), and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application such that the network application is configured to operate in accordance with a first protocol including a transmission control protocol (TCP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>[W]e’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p>https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28) </p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

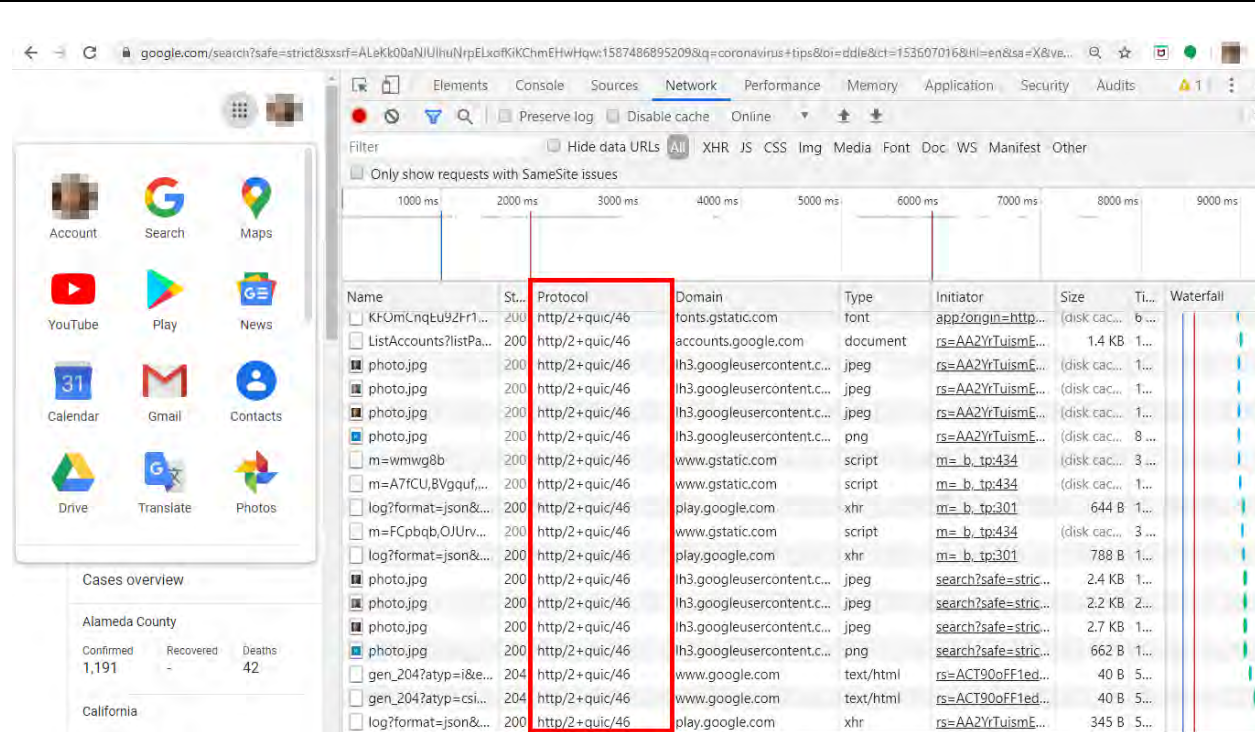
	<p>(https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p> 
<p>communicate a segment including at least one first synchronize bit;</p> <p>communicate a first acknowledgement of the segment, and at least one second synchronize bit; and</p> <p>communicate a second acknowledgement;</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) that, when operating in accordance with the first protocol to establish the TCP connection, is configured for a) communicating a segment including at least one first synchronize bit; b) communicating a first acknowledgement of the segment, and at least one second synchronize bit; and c) communicating a second acknowledgement.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a) is met by 1); b) is met by 2)/3), and c) is met by 4).</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <p>1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y</p> <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3].”</p> <p>“Request for Comments” (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
<p>wherein the network application is further configured to operate in accordance with a second protocol that is separate from the TCP, the apparatus, when operating in accordance with the second protocol to establish a second protocol connection, configured to:</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the network application is further configured to operate in accordance with a second protocol (e.g., QUIC, etc.) that is separate from the TCP. Further, the apparatus, when operating in accordance with the second protocol to establish a second protocol connection (e.g., a QUIC connection, etc.), is configured to receive idle information for detecting an idle time period, after which, the second protocol connection is subject to deactivation (e.g., closed due to inactivity, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

receive idle information for detecting an idle time period, after which, the second protocol connection is subject to deactivation;



Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfOmCnqEu92fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app/origin=hte...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	3...	
m=A7ICU,8Vgquf,...	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_tp301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_tp301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stic...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stic...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stic...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stic...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, a QUIC is separate from TCP.

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

"1. Introduction

QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.” <i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>“6. Version Negotiation</p> <p>Version negotiation ensures that client and server agree to a QUIC version that is mutually supported. A server sends a Version Negotiation packet in response to each packet that might initiate a new connection; see Section 5.2 for details.</p> <p>The size of the first packet sent by a client will determine whether a server sends a Version Negotiation packet. Clients that support multiple QUIC versions SHOULD pad the first packet they send to the largest of the minimum packet sizes across all versions they support. This ensures that the server responds if there is a mutually supported version.</p> <p>...</p> <p>6.3. Using Reserved Versions</p> <p>For a server to use a new version in the future, clients need to correctly handle unsupported versions. To help ensure this, a server SHOULD include a version that is reserved for forcing version negotiation (0x?a?a?a as defined in Section 15) when generating a Version Negotiation packet.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. A client MAY send a packet using a version that is reserved for forcing version negotiation. This can be used to solicit a list of supported versions from a server.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>“5.2.2. Server Packet Handling</p> <p>...</p> <p>If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected.</p> <p>...</p> <p>If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets.</p> <p>...</p> <p>7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p>
--	--

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,996

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	+-----+-----+-----+		
	Value	Parameter Name	Specification
	0x0000	original_connection_id	Section 18.1
	0x0001	idle_timeout	Section 18.1
	0x0002	stateless_reset_token	Section 18.1
	0x0003	max_packet_size	Section 18.1
	0x0004	initial_max_data	Section 18.1
	0x0005	initial_max_stream_data_bidi_local	Section 18.1
	0x0006	initial_max_stream_data_bidi_remote	Section 18.1
	0x0007	initial_max_stream_data_uni	Section 18.1
	0x0008	initial_max_streams_bidi	Section 18.1
	0x0009	initial_max_streams_uni	Section 18.1
	0x000a	ack_delay_exponent	Section 18.1
	0x000b	max_ack_delay	Section 18.1
	0x000c	disable_migration	Section 18.1
	0x000d	preferred_address	Section 18.1
	0x000e	active_connection_id_limit	Section 18.1
	+-----+-----+-----+		
	Table 6: Initial QUIC Transport Parameters Entries		
	QUIC: A UDP-Based Multiplexed and Secure Transport https://tools.ietf.org/html/draft-ietf-quic-transport-22		
	"18. Transport Parameter Encoding		

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>generate a second protocol packet including an idle time period parameter field identifying metadata for the idle time period based on the idle information; and</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to generate a second protocol packet (e.g., QUIC negotiation packet, etc.) including an idle time period parameter field (e.g., idle_timeout parameter field, etc.) identifying metadata (e.g., a value, etc.) for the idle time period based on the idle information.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p><u>Note</u>: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“1. Introduction</p> <p>QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].</p> <p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p><u>Note</u>: The following evidence is related to and/or describes the QUIC standard contained herein.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>“6. Version Negotiation</p> <p>Version negotiation ensures that client and server agree to a QUIC version that is mutually supported. A server sends a Version Negotiation packet in response to each packet that might initiate a new connection; see Section 5.2 for details.</p> <p>The size of the first packet sent by a client will determine whether a server sends a Version Negotiation packet. Clients that support multiple QUIC versions SHOULD pad the first packet they send to the largest of the minimum packet sizes across all versions they support. This ensures that the server responds if there is a mutually supported version.</p> <p>...</p> <p>6.3. Using Reserved Versions</p> <p>For a server to use a new version in the future, clients need to correctly handle unsupported versions. To help ensure this, a server SHOULD include a version that is reserved for forcing version negotiation (0x?a?a?a as defined in Section 15) when generating a Version Negotiation packet.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. A client MAY send a packet using a version that is reserved for forcing version negotiation. This can be used to solicit a list of supported versions from a server.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>“5.2.2. Server Packet Handling ... If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected. ... If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets. ... 7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>send, from a first node to a second node, the second protocol packet to provide the metadata for the idle time period to the second node, for use by the second node in creating or modifying, based on the metadata, a timeout attribute associated with the second protocol connection.</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to send, from a first node (e.g., the other one of the QUIC-compliant server or client, etc.) to a second node (e.g., one of a QUIC-compliant server or client, etc.), the second protocol packet (e.g., the QUIC negotiation packet, etc.) to provide the metadata (e.g., the value, etc.) for the idle time period to the second node, for use by the second node in creating or modifying, based on the metadata, a timeout attribute associated with the second protocol connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“1. Introduction</p> <p>QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].</p> <p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>“6. Version Negotiation</p> <p>Version negotiation ensures that client and server agree to a QUIC version that is mutually supported. A server sends a Version Negotiation packet in response to each packet that might initiate a new connection; see Section 5.2 for details.</p> <p>The size of the first packet sent by a client will determine whether a server sends a Version Negotiation packet. Clients that support multiple QUIC versions SHOULD pad the first packet they send to the largest of the minimum packet sizes across all versions they support. This ensures that the server responds if there is a mutually supported version.</p> <p>...</p> <p>6.3. Using Reserved Versions</p> <p>For a server to use a new version in the future, clients need to correctly handle unsupported versions. To help ensure this, a server SHOULD include a version that is reserved for forcing version negotiation (0x?a?a?a as defined in Section 15) when generating a Version Negotiation packet.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. A client MAY send a packet using a version that is reserved for forcing version negotiation. This can be used to solicit a list of supported versions from a server.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.”</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>“5.2.2. Server Packet Handling</p> <p>...</p> <p>If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected.</p> <p>...</p> <p>If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets.</p> <p>...</p> <p>7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the <code>idle_timeout</code> transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Claim 22 Elements	Applicability
<p>The apparatus of claim 15 wherein the apparatus is configured to utilize the second protocol instead of the TCP in order to permit negotiation, between the first node and the second node, of the timeout attribute, where the negotiation of the timeout attribute is not available when operating in accordance with the TCP, but is available when establishing the second protocol connection so as to permit the second protocol connection to be at least partially closed when inactive based on the timeout attribute.</p>	<p>Google infringes claim 15 and uses the apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to utilize the second protocol (e.g., QUIC, etc.) instead of the TCP in order to permit negotiation, between the first node (e.g., the other one of the QUIC-compliant server or client, etc.) and the second node (e.g., one of the QUIC-compliant server or client, etc.), of the timeout attribute, where the negotiation of the timeout attribute is not available when operating in accordance with the TCP, but is available when establishing the second protocol connection (e.g., the QUIC connection, etc.) so as to permit the second protocol connection to be at least partially closed (e.g., the connection is terminated, etc.) when inactive based on the timeout attribute.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an <code>idle_timeout</code> parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Claim 23 Elements	Applicability
<p>The apparatus of claim 15 wherein the apparatus is configured such that the timeout attribute serves to keep the second protocol connection open when inactive, and to prevent one or more other nodes from closing the second protocol connection when inactive.</p>	<p>Google infringes claim 15 and uses the apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that the timeout attribute serves to keep the second protocol connection (e.g., the QUIC connection, etc.) open when inactive, and to prevent one or more other nodes from closing the second protocol connection when inactive.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p style="padding-left: 40px;">“5.4. Required Operations on Connections</p> <p>There are certain operations which an application MUST be able to perform when interacting with the QUIC transport. This document does not specify an API, but any implementation of this version of QUIC MUST expose the ability to perform the operations described in this section on a QUIC connection.</p> <p>...</p> <p>In either role, applications need to be able to:</p> <p>...</p> <ul style="list-style-type: none"> • <u>keep a connection from silently closing, either by generating PING frames (Section 19.2) or by requesting that the transport send additional frames before the idle timeout expires (Section 10.2); and</u> • immediately close (Section 10.3) the connection.” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p style="padding-left: 40px;">“19.2. PING Frame</p> <p>Endpoints <u>can use</u> PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,996

	<p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p><u>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10).</u> However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
--	--


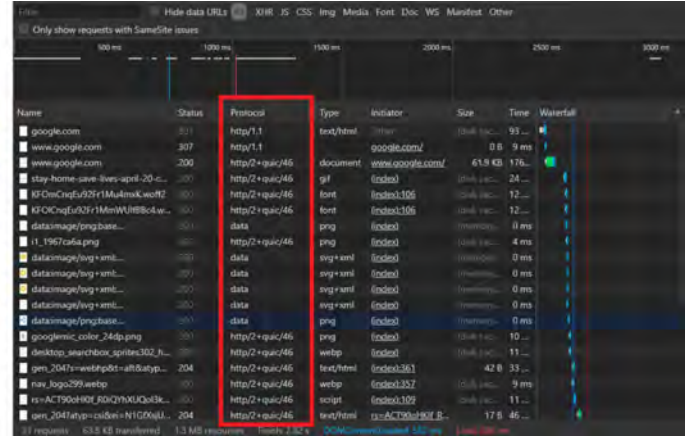
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Claim 25 Elements	Applicability
<p>An apparatus comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions such that a network application operates in accordance with a first protocol including a transmission control protocol (TCP) that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, the apparatus, when operating in accordance with the first protocol to set up a TCP connection, configured to:</p>	<p>Google uses an apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) including a non-transitory memory storing a network application (e.g., server software, etc.), and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application for carrying out the functionality set forth below.</p> <p>Note: In addition to the QUIC protocol, the apparatus also operates in accordance with the standard transmission control protocol (TCP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>[W]e’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers." https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p> https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p>Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>(https://datatracker.ietf.org/doc/html/draft-ietf-quick-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quick-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quick-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quick-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quick-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quick-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quick-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p> <div data-bbox="808 662 1197 987">  </div> <div data-bbox="1213 662 1894 1096">  </div>
<p>communicate a segment including at least one first synchronize bit;</p> <p>communicate a first acknowledgement of the segment, and at least one second synchronize bit; and</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) that, when operating in accordance with the first protocol to establish the TCP connection, is configured for a) communicating a segment including at least one first synchronize bit; b) communicating a first acknowledgement of the segment, and at least one second synchronize bit; and c) communicating a second acknowledgement.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

communicate a second acknowledgement;	<p>Note: As set forth below, a) is met by 1); b) is met by 2)/3), and c) is met by 4).</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <p>1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y</p> <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3].”</p> <p>“Request for Comments” (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
receive, by a second node from a first node, a packet;	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to receive, by a second node (e.g., one of a QUIC-compliant server or client, etc.) from a first node (e.g., the other one of the QUIC-compliant server or client, etc.), a packet (e.g., QUIC negotiation packet, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“6. Version Negotiation</p> <p>Version negotiation ensures that client and server agree to a QUIC version that is mutually supported. A server sends a Version Negotiation packet in response to each packet that might initiate a new connection; see Section 5.2 for details.</p> <p>The size of the first packet sent by a client will determine whether a server sends a Version Negotiation packet. Clients that support multiple QUIC versions SHOULD pad the first packet they send to the largest of the minimum packet sizes across all versions they support. This ensures that the server responds if there is a mutually supported version.</p> <p>...</p> <p>6.3. Using Reserved Versions</p> <p>For a server to use a new version in the future, clients need to correctly handle unsupported versions. To help ensure this, a server SHOULD include a version that is reserved for forcing version negotiation (0x?a?a?a as defined in Section 15) when generating a Version Negotiation packet.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. A client MAY send a packet using a version that is reserved for forcing version negotiation. This can be used to solicit a list of supported versions from a server.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

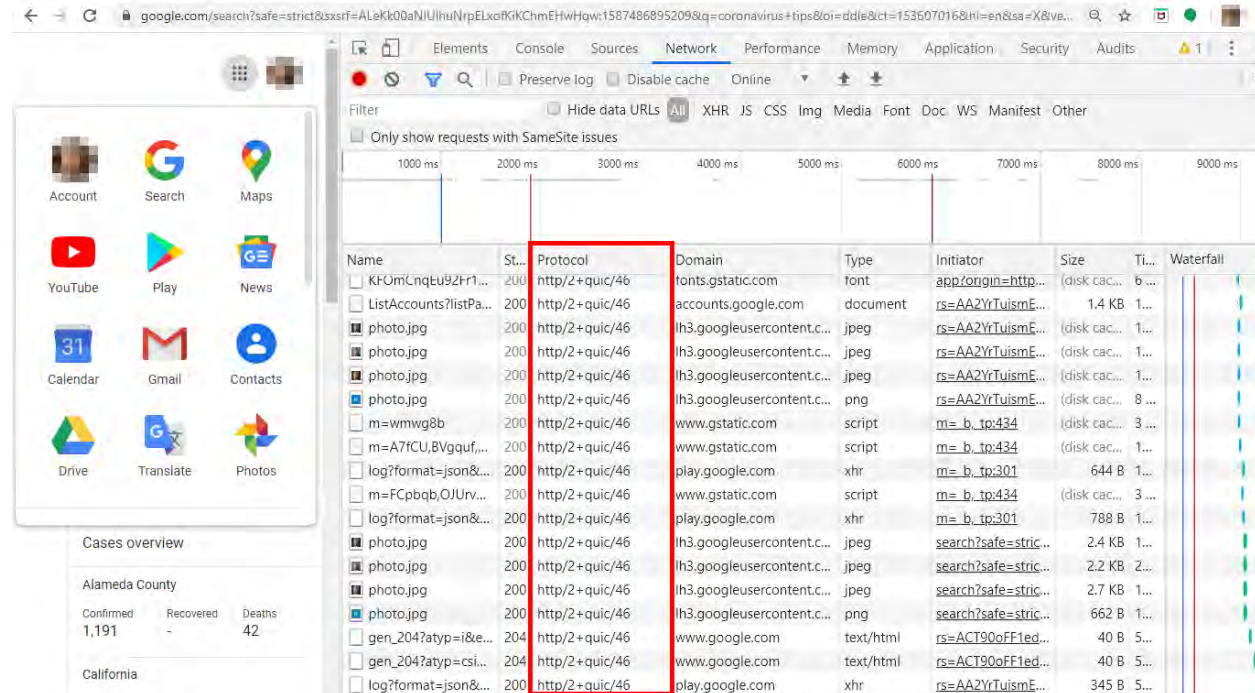
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>“5.2.2. Server Packet Handling</p> <p>...</p> <p>If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected.</p> <p>...</p> <p>If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets.</p> <p>...</p> <p>7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
wherein the network application is further configured to operate in accordance with a second protocol that is different from the TCP, where the second protocol operates above the IP layer and below the HTTP application layer, the apparatus, when operating in accordance with the second	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the network application is further configured to operate in accordance with a second protocol (e.g., QUIC, etc.) that is different from the TCP, where the second protocol operates above the IP layer and below the HTTP application layer, the apparatus, when operating in accordance with the second protocol to set up a second protocol connection (e.g., a QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

protocol to set up a second protocol connection, configured to:

Note: Below is a web page of Google (<https://www.google.com/>).



Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfOmCnqEu92Fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app/origin=http...			
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...		1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...		1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...		1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...		8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m=b_ip434		3...	
m=A7fCU,BVgquf,...	200	http/2+quic/46	www.gstatic.com	script	m=b_ip434		1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_ip301	644 B	1...	
m=FCpbqb,OJUnv...	200	http/2+quic/46	www.gstatic.com	script	m=b_ip434		3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_ip301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, a QUIC is different from TCP.

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

"1. Introduction

QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.” <i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>“6. Version Negotiation</p> <p>Version negotiation ensures that client and server agree to a QUIC version that is mutually supported. A server sends a Version Negotiation packet in response to each packet that might initiate a new connection; see Section 5.2 for details.</p> <p>The size of the first packet sent by a client will determine whether a server sends a Version Negotiation packet. Clients that support multiple QUIC versions SHOULD pad the first packet they send to the largest of the minimum packet sizes across all versions they support. This ensures that the server responds if there is a mutually supported version.</p> <p>...</p> <p>6.3. Using Reserved Versions</p> <p>For a server to use a new version in the future, clients need to correctly handle unsupported versions. To help ensure this, a server SHOULD include a version that is reserved for forcing version negotiation (0x?a?a?a as defined in Section 15) when generating a Version Negotiation packet.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. A client MAY send a packet using a version that is reserved for forcing version negotiation. This can be used to solicit a list of supported versions from a server.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>“5.2.2. Server Packet Handling</p> <p>...</p> <p>If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected.</p> <p>...</p> <p>If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets.</p> <p>...</p> <p>7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>identify metadata, that specifies at least one of a number of seconds or minutes, in an idle time period parameter field in the packet for an idle time period and, during which, no packet is communicated that meets each of the following criteria: a) communicated via the second protocol connection, and b) causes the second protocol connection to be kept at least partially alive; and</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to identify metadata (e.g., a value, etc.), that specifies at least one of a number of seconds or minutes, in an idle time period parameter field (e.g., idle_timeout parameter field, etc.) in the packet for an idle time period and, during which, no packet is communicated that meets each of the following criteria: a) communicated via the second protocol connection (e.g., the QUIC connection, etc.), and b) causes the second protocol connection to be kept at least partially alive.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>[QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
determine, by the second node and based on the metadata, a timeout attribute associated with the second protocol connection.	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to determine, by the second node (e.g., one of the QUIC-compliant server or client, etc.) and based on the metadata (e.g., the value, etc.), a timeout attribute associated with the second protocol connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily determined based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Claim 27 Elements	Applicability
<p>The apparatus of claim 25 wherein the apparatus is configured for: detecting the idle time period based on the timeout attribute; and deactivating the second protocol connection by communicating a particular packet between the first node and the second node, in response to detecting the idle time period.</p>	<p>Google infringes claim 25 and uses the apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to detect the idle time period based on the timeout attribute and, in response thereto, deactivate the second protocol connection (e.g., terminate the connection, etc.) by communicating a particular packet between the first node (e.g., the other one of the QUIC-compliant server or client, etc.) and the second node (e.g., one of the QUIC-compliant server or client, etc.), in response to detecting the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,996

	<p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--


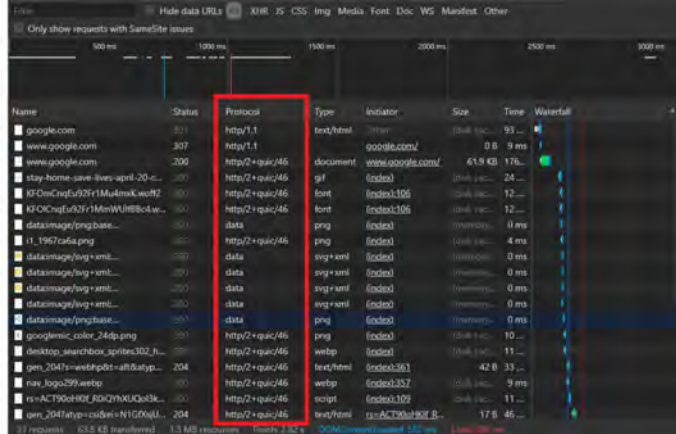
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Claim 29 Elements	Applicability
<p>An apparatus comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions such that a network application operates in accordance with a first protocol including a transmission control protocol (TCP) that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, the apparatus, when operating in accordance with the first protocol to set up a TCP connection, configured to:</p>	<p>Google uses an apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) including a non-transitory memory storing a network application (e.g., server software, etc.), and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application for carrying out the functionality set forth below.</p> <p>Note: In addition to the QUIC protocol, the apparatus also operates in accordance with the standard transmission control protocol (TCP).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>“Google Cloud Platform Blog...</p> <p>[W]e’re happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers.” https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p> https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p> <div style="display: flex; justify-content: space-around; align-items: center;">   </div>
<p>communicate a segment including at least one first synchronize bit;</p> <p>communicate a first acknowledgement of the</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to a) communicate a segment including at least one first synchronize bit; b) communicate a first acknowledgement of the segment, and at least one second synchronize bit; and c) communicate a second acknowledgement.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

<p>segment, and at least one second synchronize bit; and</p> <p>communicate a second acknowledgement;</p>	<p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a) is met by 1); b) is met by 2)/3), and c) is met by 4).</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <p>1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y</p> <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN. The three way handshake and the advantages of a clock-driven scheme are discussed in [3].”</p> <p>“Request for Comments” (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
<p>wherein the network application is further configured to operate in accordance with a second protocol that is different from the TCP, where the second protocol operates above the IP layer and</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the network application is further configured to operate in accordance with a second protocol (e.g., QUIC, etc.) that is different from the TCP, where the second protocol operates above the IP layer and below the HTTP application layer, the apparatus, when operating in accordance with the second protocol to set up a second protocol connection (e.g., a QUIC connection, etc.).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

below the HTTP application layer, the apparatus, when operating in accordance with the second protocol to set up a second protocol connection, configured to:

See excerpt(s) below, for example (emphasis added, if any):

Note: Below is a web page of Google (<https://www.google.com/>).

Name	St...	Protocol	Domain	Type	Initiator	Size	Ti...	Waterfall
KfOmCnqEu92fr1...	200	http/2+quic/46	fonts.gstatic.com	font	app/origin=http...	(disk cac...	6...	
ListAccounts?listPa...	200	http/2+quic/46	accounts.google.com	document	rs=AA2YrTuismE...	1.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	rs=AA2YrTuismE...	(disk cac...	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	rs=AA2YrTuismE...	(disk cac...	8...	
m=wmwg8b	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	3...	
m=A7fCU,BVgquf...	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	1...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_tp301	644 B	1...	
m=FCpbqb,OJUr...	200	http/2+quic/46	www.gstatic.com	script	m=b_tp434	(disk cac...	3...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	m=b_tp301	788 B	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.4 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.2 KB	2...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	jpeg	search?safe=stria...	2.7 KB	1...	
photo.jpg	200	http/2+quic/46	lh3.googleusercontent.c...	png	search?safe=stria...	662 B	1...	
gen_204?atyp=i&e...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
gen_204?atyp=csi...	204	http/2+quic/46	www.google.com	text/html	rs=ACT90oFF1ed...	40 B	5...	
log?format=json&...	200	http/2+quic/46	play.google.com	xhr	rs=AA2YrTuismE...	345 B	5...	

Note: As set forth below, a QUIC is different from TCP.

Note: The following evidence is related to and/or describes the QUIC standard contained herein.

“1. Introduction

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].</p> <p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.” <i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p><u>Note</u>: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p><u>Note</u>: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>“6. Version Negotiation</p> <p>Version negotiation ensures that client and server agree to a QUIC version that is mutually supported. A server sends a Version Negotiation packet in response to each packet that might initiate a new connection; see Section 5.2 for details.</p> <p>The size of the first packet sent by a client will determine whether a server sends a Version Negotiation packet. Clients that support multiple QUIC versions SHOULD pad the first packet they send to the largest of the minimum packet sizes across all versions they support. This ensures that the server responds if there is a mutually supported version.</p> <p>...</p> <p>6.3. Using Reserved Versions</p> <p>For a server to use a new version in the future, clients need to correctly handle unsupported versions. To help ensure this, a server SHOULD include a version that is reserved for forcing version negotiation (0x?a?a?a as defined in Section 15) when generating a Version Negotiation packet.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. A client MAY send a packet using a version that is reserved for forcing version negotiation. This can be used to solicit a list of supported versions from a server.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>“5.2.2. Server Packet Handling ... If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected. ... If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets. ...</p> <p>7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>receive idle information for use in detecting an idle time period during which no signal is communicated that meets each of the following criteria: a) communicated in the second protocol connection, and b) results in the second protocol connection being at least partially kept alive;</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to receive idle information for use in detecting an idle time period during which no signal is communicated that meets each of the following criteria: a) communicated in the second protocol connection (e.g., the QUIC connection, etc.), and b) results in the second protocol connection being at least partially kept alive.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“1. Introduction</p> <p>QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].</p> <p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,996

	<p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, after the idle time period is detected, the QUIC connection is closed due to inactivity.</p> <p>“10.2. Idle Timeout</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO)."</p> <p>"The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>"19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows."</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

<p>generate, based on the idle information, a second protocol packet including an idle time period parameter field identifying metadata that is specified in at least one of a number of seconds or minutes; and</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to generate, based on the idle information, a second protocol packet (e.g., QUIC negotiation packet, etc.) including an idle time period parameter field (e.g., idle_timeout parameter field, etc.) identifying metadata (e.g., a value, etc.) that is specified in at least one of a number of seconds or minutes.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p><u>Note</u>: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“1. Introduction</p> <p>QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].</p> <p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p><u>Note</u>: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p><u>Note</u>: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>“6. Version Negotiation</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>Version negotiation ensures that client and server agree to a QUIC version that is mutually supported. A server sends a Version Negotiation packet in response to each packet that might initiate a new connection; see Section 5.2 for details.</p> <p>The size of the first packet sent by a client will determine whether a server sends a Version Negotiation packet. Clients that support multiple QUIC versions SHOULD pad the first packet they send to the largest of the minimum packet sizes across all versions they support. This ensures that the server responds if there is a mutually supported version.</p> <p>...</p> <p>6.3. Using Reserved Versions</p> <p>For a server to use a new version in the future, clients need to correctly handle unsupported versions. To help ensure this, a server SHOULD include a version that is reserved for forcing version negotiation (0x?a?a?a as defined in Section 15) when generating a Version Negotiation packet.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. A client MAY send a packet using a version that is reserved for forcing version negotiation. This can be used to solicit a list of supported versions from a server.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“5.2.2. Server Packet Handling</p> <p>...</p> <p>If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected.</p> <p>...</p> <p>If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets.</p> <p>...</p> <p>7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quick-transport-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quick-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p>
--	---

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>send, from a first node to a second node and during the set up of the second protocol connection, the second protocol packet to provide the metadata to the second node, for use by the second node in determining a timeout attribute associated with the second protocol connection.</p>	<p>Google uses the apparatus (e.g., the one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to send, from a first node (e.g., the other one of the QUIC-compliant server or client, etc.) to a second node (e.g., one of a QUIC-compliant server or client, etc.) and during the set up of the second protocol connection, the second protocol packet (e.g., the QUIC negotiation packet, etc.) to provide the metadata (e.g., the value, etc.) to the second node, for use by the second node in determining a timeout attribute associated with the second protocol connection (e.g., the QUIC connection, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p> <p>“1. Introduction</p> <p>QUIC is a new multiplexed and secure transport atop UDP. QUIC builds on decades of transport and security experience, and implements mechanisms that make it attractive as a modern general-purpose transport. The QUIC protocol is described in [QUIC-TRANSPORT].</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>QUIC implements the spirit of existing TCP loss recovery mechanisms, described in RFCs, various Internet-drafts, and also those prevalent in the Linux TCP implementation. This document describes QUIC congestion control and loss recovery, and where applicable, attributes the TCP equivalent in RFCs, Internet-drafts, academic papers, and/or TCP implementations.”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://tools.ietf.org/html/draft-ietf-quic-recovery-22</p> <p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since QUIC is built on top of UDP, it suffers from no such limitations.” https://www.chromium.org/quic</p> <p>Note: As set forth below, a QUIC connection establishment begins by sending a handshake packet.</p> <p>“6. Version Negotiation</p> <p>Version negotiation ensures that client and server agree to a QUIC version that is mutually supported. A server sends a Version Negotiation packet in response to each packet that might initiate a new connection; see Section 5.2 for details.</p> <p>The size of the first packet sent by a client will determine whether a server sends a Version Negotiation packet. Clients that support multiple QUIC versions SHOULD pad the first packet they send to the largest of the minimum packet sizes across all versions they support. This ensures that the server responds if there is a mutually supported version.</p> <p>...</p> <p>6.3. Using Reserved Versions</p> <p>For a server to use a new version in the future, clients need to correctly handle unsupported versions. To help ensure this, a server SHOULD include a version that is reserved for forcing version negotiation (0x?a?a?a?a as defined in Section 15) when generating a Version Negotiation packet.</p> <p>The design of version negotiation permits a server to avoid maintaining state for packets that it rejects in this fashion. A client MAY send a packet using a version that is reserved for forcing version negotiation. This can be used to solicit a list of supported versions from a server.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“The initial handshake packet from the client needs to fit in a single packet (#338)”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC packet: A complete processable unit of QUIC that can be encapsulated in a UDP datagram. Multiple QUIC packets can be encapsulated in a single UDP datagram.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, prior to a QUIC connection being established, the QUIC connection is “set up” using the aforementioned handshake.</p> <p>“5.2.2. Server Packet Handling</p> <p>...</p> <p>If the packet is an Initial packet fully conforming with the specification, the server proceeds with the handshake (Section 7). This commits the server to the version that the client selected.</p> <p>...</p> <p>If the packet is a 0-RTT packet, the server MAY buffer a limited number of these packets in anticipation of a late-arriving Initial packet. Clients are not able to send Handshake packets prior to receiving a server response, so servers SHOULD ignore any such packets.</p> <p>...</p> <p>7. Cryptographic and Transport Handshake</p> <p>QUIC relies on a combined cryptographic and transport handshake to minimize connection establishment latency. QUIC uses the CRYPTO frame Section 19.6 to transmit the cryptographic handshake. Version 0x00000001 of QUIC uses TLS as described in [QUIC-TLS]; a different QUIC version number could indicate that a different cryptographic handshake protocol is in use.</p> <p>QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, the metadata includes a value in seconds that is encoded as an unsigned 16-bit integer.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p style="padding-left: 40px;">Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p style="padding-left: 80px;">The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p style="padding-left: 40px;">The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p style="padding-left: 40px;">A connection will time out if no packets are sent or received for a period longer than the time specified in the idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,996

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Claim 30 Elements	Applicability
<p>The apparatus of claim 29 wherein the apparatus is configured such that: the determination of the timeout attribute results from a negotiation between the first node and the second node;</p>	<p>Google infringes claim 29 and uses the apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is configured such that: the determination of the timeout attribute results from a negotiation between the first node (e.g., the other one of the QUIC-compliant server or client, etc.) and the second node (e.g., one of the QUIC-compliant server or client, etc.).</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled by either peer, a connection is silently closed and its state is discarded when it remains idle for longer than the minimum of the max_idle_timeouts (see Section 18.2) and three times the current Probe Timeout (PTO).</p> <p><u>Each endpoint advertises a max_idle_timeout, but the effective value at an endpoint is computed as the minimum of the two advertised values.</u> By announcing a max_idle_timeout, an endpoint commits to initiating an immediate close (Section 10.3) if it abandons the connection prior to the effective value.</p> <p>An endpoint restarts its idle timer when a packet from its peer is received and processed successfully. The idle timer is also restarted when sending an ack-eliciting packet (see [QUIC-RECOVERY]), but only if no other ack-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity. An endpoint might need to send packets to avoid an idle timeout if it is unable to send application data due to being blocked on flow control limits; see Section 4.</p>
--	--

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>An endpoint that sends packets near the end of the idle timeout period risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could time out within a Probe Timeout (PTO; see Section 6.6 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
<p>wherein the apparatus is further configured for: detecting the idle time period based on the idle information; and</p>	<p>Google infringes claim 29 and uses the apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) wherein the apparatus is further configured for: detecting the idle time period based on the idle information.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

Value	Parameter Name	Specification
0x0000	original_connection_id	Section 18.1
0x0001	idle_timeout	Section 18.1
0x0002	stateless_reset_token	Section 18.1
0x0003	max_packet_size	Section 18.1
0x0004	initial_max_data	Section 18.1
0x0005	initial_max_stream_data_bidi_local	Section 18.1
0x0006	initial_max_stream_data_bidi_remote	Section 18.1
0x0007	initial_max_stream_data_uni	Section 18.1
0x0008	initial_max_streams_bidi	Section 18.1
0x0009	initial_max_streams_uni	Section 18.1
0x000a	ack_delay_exponent	Section 18.1
0x000b	max_ack_delay	Section 18.1
0x000c	disable_migration	Section 18.1
0x000d	preferred_address	Section 18.1
0x000e	active_connection_id_limit	Section 18.1

Table 6: Initial QUIC Transport Parameters Entries

QUIC: A UDP-Based Multiplexed and Secure Transport <https://tools.ietf.org/html/draft-ietf-quic-transport-22>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
<p>releasing a resource allocated for the second protocol connection by one of the first and second nodes without signaling another one of the first and second nodes in connection with the detection of the idle time period.</p>	<p>Google infringes claim 29 and uses the apparatus (e.g., one or more servers and/or the Google Cloud Platform, or at least one component thereof, etc.) to release a resource allocated for the second protocol connection (e.g., the QUIC connection, etc.) by one of the first and second nodes without signaling another one of the first and second nodes in connection with the detection of the idle time period.</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 9,923,996

	<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p>“The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
--	---

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 9,923,996

Caveat: The notes and/or cited excerpts utilized herein are set forth for illustrative purposes only and are not meant to be limiting in any manner. For example, the notes and/or cited excerpts, may or may not be supplemented or substituted with different excerpt(s) of the relevant reference(s), as appropriate. Further, to the extent any error(s) and/or omission(s) exist herein, all rights are reserved to correct the same.

JENAM TECH, LLC'S FIRST AMENDED SET OF INFRINGEMENT CONTENTIONS

U.S. Patent No. 10,951,742 – Google LLC

Claims 1-3, 6, 10, 12, 15, 20-21, 27-28, 35, 39, 48, 51-52, 55-56, 63-65, 67-69, 71-78, 80, 83-84, 86, 88, 103-104, 159, 167-176

Jenam Tech LLC (“Jenam”) provides evidence of infringement of claims 1-3, 6, 10, 12, 15, 20-21, 27-28, 35, 39, 48, 51-52, 55-56, 63-65, 67-69, 71-78, 80, 83-84, 86, 88, 103-104, 159, 167-176 of U.S. Patent No. 10,951,742 (hereinafter “the ’742 patent”) by Google LLC (“Google”). In support thereof, Jenam provides the following claim charts.

“Accused Instrumentalities” as used herein refers to at least one or more websites or web addresses including, but not limited to www.google.com, stored and/or hosted on one or more servers owned or under the control of Google. For the sake of clarity, Jenam alleges that all of Google’s various websites, products, and platforms utilizing QUIC infringe the ’742 patent, including, but not limited to Google Edge Network, Google Cloud, Chrome Enterprise, G suite, Google Play, Chrome, Android (Android Enterprise, Android Messages (RCS)), Duo, Google Ads, Adwords, Google Analytics, Youtube, Google Mobile apps, Google shopping, and Google Maps. A list of Google’s “products” can be found at https://about.google/intl/en_us/products/. On information and belief, the Accused Instrumentalities referred to in the charts below are representative of Google’s use of QUIC in Google’s other websites, products and platforms. These claim charts demonstrate Google’s infringement, and provide notice of such infringement, by comparing each element of the asserted claims to corresponding components, aspects, and/or features of the Accused Instrumentalities. These claim charts are not intended to constitute an expert report on infringement. These claim charts include information provided by way of example, and not by way of limitation.

The analysis set forth below is based only upon information from publicly available resources regarding the Infringing Instrumentalities. An analysis of Google’s (or other third parties’) technical documentation and/or software source code may assist in fully identify all infringing features and functionality. Accordingly, Jenam reserves the right to supplement this infringement analysis once such information is made available to Jenam. Furthermore, Jenam reserves the right to revise this infringement analysis, as appropriate, upon issuance of a court order construing any terms recited in the asserted claims.

Unless otherwise noted, Jenam contends that Google directly infringes the ’742 patent in violation of 35 U.S.C. § 271(a) by selling, offering to sell, making, using, and/or importing the Infringing Instrumentalities. The following exemplary analysis demonstrates that infringement. Unless otherwise noted, Jenam further contends that the evidence below supports a finding of indirect infringement under 35 U.S.C. §§ 271(b) and/or (c), in conjunction with other evidence of liability under one or more of those subsections. Google makes, uses, sells, imports, or offers for sale in the United States, or has made, used, sold, imported, or offered for sale in the past, without authority, or induces others to make, use, sell, import, or offer for sale in the United States, or has induced others to make, use, sell, import, or offer for sale in the past, without authority products, equipment, or services that infringe claims 1-3, 6, 10, 12, 15, 20-21, 27-28, 35, 39, 48, 51-52, 55-56, 63-65, 67-69, 71-78, 80, 83-84, 86, 88, 103-104, 159, 167-176 of the ’742 patent, including without limitation, the Accused Instrumentalities.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Unless otherwise noted, Jenam believes and contends that each element of each claim asserted herein is literally met through Google's provision of the Infringing Instrumentalities. However, to the extent that Google attempts to allege that any asserted claim element is not literally met, Jenam believes and contends that such elements are met under the doctrine of equivalents. More specifically, in its investigation and analysis of the Infringing Instrumentalities, Jenam did not identify any substantial differences between the elements of the patent claims and the corresponding features of the Infringing Instrumentalities, as set forth herein. In each instance, the identified feature of the Infringing Instrumentalities performs at least substantially the same function in substantially the same way to achieve substantially the same result as the corresponding claim element.

To the extent the chart of an asserted claim relies on evidence about certain specifically-identified Accused Instrumentalities, Jenam asserts that, on information and belief, any similarly-functioning instrumentalities also infringes the charted claim. Jenam reserves the right to amend this infringement analysis based on other products made, used, sold, imported, or offered for sale by Google. Jenam also reserves the right to amend this infringement analysis by citing other claims of the '742 patent, not listed in the claim chart, that are infringed by the Accused Instrumentalities. Jenam further reserves the right to amend this infringement analysis by adding, subtracting, or otherwise modifying content in the "Accused Instrumentalities" column of each chart.

Claim	Claim Elements	Applicability
1	An apparatus, comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions to:	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.), comprising: a non-transitory memory storing instructions (e.g., server software, etc.), and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions to:</p> <p>See excerpt(s) below, for example (emphasis added, if any):</p> <p>Note: Google admits that their "Google Cloud Platform" uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>we're happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers."</p> <p>https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p>

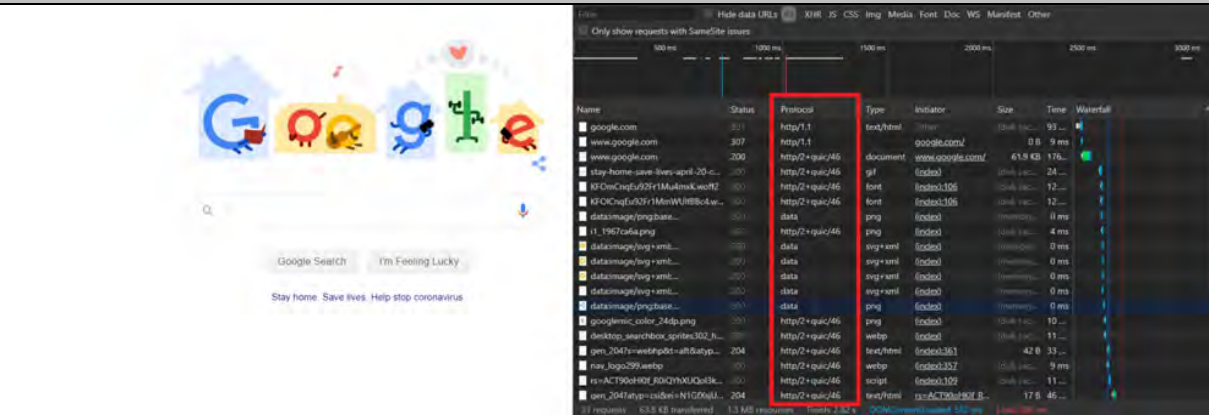
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p> https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 </p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p> https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities. </p> <p> Note: Below is a web page of Google (https://www.google.com/). </p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		
	<p>identify, at a first node, first information on which at least a first duration for detecting a first type of time period is based;</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.), configured to identify, at a first node (e.g., one of a QUIC-compliant server or client, etc.), first information (e.g., first value, etc.) on which at least a first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) for detecting a first type of time period (e.g., idle-type timeout, etc.) is based;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the idle timeout field of the connection negotiation packet.</u></p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“3.3. Session resumption versus Keep-alive</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p> <ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods. • <u>Resume the session after a long idle period, using 0-RTT resumption when appropriate.</u> <p>...</p> <p>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic. In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. <u>When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with restarting the connection.</u> This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.”</p> <p>https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
	allocate a first resource for a first non-Transmission Control Protocol (non-TCP) connection;	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.), configured to allocate a first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) for a first non-Transmission Control Protocol (non-TCP) connection (e.g., QUIC connection, etc.);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“Google Cloud Platform (GCP), offered by Google (company), is a <u>suite of cloud computing services</u> that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>1. Introduction</p> <p>QUIC is a multiplexed and secure transport protocol that runs on top of UDP. QUIC aims to provide a flexible set of features that allow it to be a general-purpose transport for multiple applications.</p> <p>QUIC implements techniques learned from experience with TCP, SCTP and <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quick-transport-07</p>
	generate a first non-TCP packet including a first parameter field identifying first metadata for use in determining a second duration for detecting the first type of time period;	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.), configured to generate a first non-TCP packet (e.g., first QUIC negotiation packet, etc.) including a first parameter field (e.g., first idle_timeout parameter field, etc.) identifying first metadata (e.g., first idle timeout value, etc.) for use in determining a second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) for detecting the first type of time period (e.g., idle-type timeout, etc.);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<p>are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p> <p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;”</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	set up the first non-TCP connection, by sending, from the first node to a	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.), configured to set up the first non-TCP connection (e.g., QUIC connection, etc.), by sending, from the first node (e.g., one of a QUIC-compliant server or client, etc.) to a</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	second node, the first non-TCP packet to provide the first metadata to the second node, for use by the second node in determining the second duration for detecting the first type of time period;	<p>second node (e.g., the other one of the QUIC-compliant server or client, etc.), the first non-TCP packet (e.g., first QUIC negotiation packet, etc.) to provide the first metadata (e.g., first idle timeout value, etc.) to the second node, for use by the second node in determining the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) for detecting the first type of time period (e.g., idle-type timeout, etc.);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> authenticated key exchange, where <ul style="list-style-type: none"> a server is always authenticated, a client is optionally authenticated, every connection produces distinct and unrelated keys, keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and 1-RTT keys have forward secrecy <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>in response to detecting, based on the first duration and by the first node during at least a portion of the first non-TCP connection including at least a portion of the first non-TCP connection set up, a first time period of the first type of time period, at least partially close the first non-TCP connection, by releasing, by the first node, the first</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.), configured to, in response to detecting, based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) and by the first node (e.g., one of a QUIC-compliant server or client, etc.) during at least a portion of the first non-TCP connection (e.g., QUIC connection, etc.) including at least a portion of the first non-TCP connection set up, a first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.), at least partially close (e.g., terminate, etc.) the first non-TCP connection, by releasing, by the first node, the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/rcv buffers, etc.) allocated for the first non-TCP connection;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <u>idle timeout value sets the duration of idleness, after which the connection is shutdown</u>, a timeout attribute of the connection is necessarily</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	resource allocated for the first non-TCP connection;	<p>detected based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC <u>connection can be terminated</u> in one of three ways:</p> <ul style="list-style-type: none"> • <u>idle timeout</u> (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	in response to detecting, based on the second duration and by the first node after the first duration is changed to the second duration, a second	Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.), configured to, in response to detecting, based on the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) and by the first node (e.g., one of a QUIC-compliant server or client, etc.) after the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is changed to the second duration, a second time period (e.g., idle timeout in connection with 1-RTT

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	time period of the first type of time period, at least partially close the first non-TCP connection, by releasing, by the first node, the first resource allocated for the first non-TCP connection; and	<p>packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.), at least partially close (e.g., terminate, etc.) the first non-TCP connection (e.g., QUIC connection, etc.), by releasing, by the first node, the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/rcv buffers, etc.) allocated for the first non-TCP connection; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: <code>original_connection_id</code>, <code>preferred_address</code>, <code>stateless_reset_token</code>, <code>ack_delay_exponent</code> and <code>active_connection_id_limit</code>. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • <code>initial_max_data</code> • <code>initial_max_stream_data_bidi_local</code> • <code>initial_max_stream_data_bidi_remote</code> • <code>initial_max_stream_data_uni</code> • <code>initial_max_streams_bidi</code> • <code>initial_max_streams_uni</code> <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes <code>initial_max_data</code> and either <code>initial_max_streams_bidi</code> and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>in response to detecting, based on a third duration, a third time period, at least partially close the first non-TCP connection, by releasing, by the first node, the first resource allocated for the first non-TCP connection, where the third duration is determined based on a first algorithm that is different from a second algorithm on which a determination</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.), configured to, in response to detecting, based on a third duration (e.g., duration associated with probe timeout, etc.), a third time period (e.g., probe timeout, etc.), at least partially close (e.g., terminate, etc.) the first non-TCP connection (e.g., QUIC connection, etc.), by releasing, by the first node (e.g., one of a QUIC-compliant server or client, etc.), the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/rcv buffers, etc.) allocated for the first non-TCP connection, where the third duration is determined based on a first algorithm (e.g., used to determine the probe timeout, etc.) that is different from a second algorithm (e.g., used to negotiate an idle timeout period between nodes, etc.) on which a determination of the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is based.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	of the second duration is based.	<p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p>
2	The apparatus of claim 1, wherein the apparatus is configured such that the first duration is not negotiated between the first and second nodes during the first non-TCP connection set up, and the second duration is negotiated between the first and second nodes during the first non-TCP connection set up for the first non-TCP connection.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is not negotiated between the first and second nodes during the first non-TCP connection (e.g., QUIC connection, etc.) set up, and the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is negotiated between the first and second nodes during the first non-TCP connection set up for the first non-TCP connection.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
3	The apparatus of claim 1, wherein the apparatus is configured such that the first time period is only capable of being detected	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) is only capable of being detected during the first non-TCP connection (e.g., QUIC connection, etc.) set up.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	during the first non-TCP connection set up.	<p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
6	<p>The apparatus of claim 1, wherein the apparatus is configured such that the third time period is detected based on the third duration, by being detected when no acknowledgement packet is received by the first node in the first non-TCP connection and processed as an acknowledgement, during the third duration, for a sent packet sent by the first node in the first non-TCP connection.</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that the third time period (e.g., probe timeout, etc.) is detected based on the third duration (e.g., duration associated with probe timeout, etc.), by being detected when no acknowledgement packet is received by the first node (e.g., one of a QUIC-compliant server or client, etc.) in the first non-TCP connection (e.g., QUIC connection, etc.) and processed as an acknowledgement, during the third duration, for a sent packet sent by the first node in the first non-TCP connection.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> • <code>max_ack_delay</code> (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a <code>max_ack_delay</code> of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
10	The apparatus of claim 1, wherein the apparatus is	Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that the first non-TCP packet (e.g.,

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	configured such that the first non-TCP packet also includes data, separate from the first metadata, for use in determining the third time period.	<p>first QUIC negotiation packet, etc.) also includes data (e.g. metadata exchanged for setting max_ack_delay, etc.), separate from the first metadata (e.g., first idle timeout value, etc.), for use in determining the third time period (e.g., probe timeout, etc.).</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>$PTO = \text{smoothed_rtt} + \max(4 * \text{rttvar}, k\text{Granularity}) + \text{max_ack_delay}$</p> <p>$k\text{Granularity}$, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate ($4 * \text{rttvar}$), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least $k\text{Granularity}$, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
12	The apparatus of claim 1, wherein the apparatus is configured such that the first duration is not negotiated between the first and second nodes during the first non-TCP connection set up, the second duration is negotiated during the first non-TCP connection set up between the first and second nodes for the first	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is not negotiated between the first and second nodes during the first non-TCP connection (e.g., QUIC connection, etc.) set up, the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is negotiated during the first non-TCP connection set up between the first and second nodes for the first non-TCP connection, and the third duration (e.g., duration associated with probe timeout, etc.) of the third time period (e.g., probe timeout, etc.) is negotiated during the first non-TCP connection set up between the first and second nodes for the first non-TCP connection (e.g. the third time period is calculated based on the max_ack_delay parameter, etc.).</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	non-TCP connection, and the third duration of the third time period is negotiated during the first non-TCP connection set up between the first and second nodes for the first non-TCP connection.	<p>Note: As set forth below, a <u>QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient</u> of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>packets</u>. Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: <code>original_connection_id</code>, <code>preferred_address</code>, <code>stateless_reset_token</code>, <code>ack_delay_exponent</code> and <code>active_connection_id_limit</code>. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • <code>initial_max_data</code> • <code>initial_max_stream_data_bidi_local</code> • <code>initial_max_stream_data_bidi_remote</code> • <code>initial_max_stream_data_uni</code> • <code>initial_max_streams_bidi</code> • <code>initial_max_streams_uni</code>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
15	The apparatus of claim 1, wherein the apparatus is configured such that the third time period is a second type of time period.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that the third time period (e.g., probe timeout, etc.) is a second type of time period (e.g., probe-type timeout, etc.).</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
20	The apparatus of claim 1, wherein the apparatus is configured such that the third duration is determined using a second parameter field identifying second metadata, where the second parameter field is communicated in the first non-TCP connection.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that the third duration (e.g., duration associated with probe timeout, etc.) is determined using a second parameter field (e.g., second idle_timeout parameter field, etc.) identifying second metadata (e.g., second idle timeout value, etc.), where the second parameter field is communicated in the first non-TCP connection (e.g., QUIC connection, etc.).</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
21	The apparatus of claim 20, wherein the apparatus is configured such that the second duration is determined using the first parameter field identifying the first metadata, and the second parameter field identifying the second metadata.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 20 and is configured such that the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is determined using the first parameter field (e.g., first idle_timeout parameter field, etc.) identifying the first metadata (e.g., first idle timeout value, etc.), and the second parameter field (e.g., second idle_timeout parameter field, etc.) identifying the second metadata (e.g., second idle timeout value, etc.).</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, k\text{Granularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
27	The apparatus of claim 1, wherein the apparatus is configured such that the first type of time period includes an idle time period.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that the first type of time period (e.g., idle-type timeout, etc.) includes an idle time period.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
28	The apparatus of claim 27, wherein the apparatus is configured such that the idle time period is one: during which, no packet is communicated in the non-TCP connection to keep the non-TCP connection active.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 27 and is configured such that the idle time period is one: during which, no packet is communicated in the non-TCP connection (e.g., QUIC connection, etc.) to keep the non-TCP connection active.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-27.txt</p>
35	The apparatus of claim 28, wherein the apparatus is configured such that the idle time period is one: during which, no packet is sent or received by the apparatus in the non-TCP connection to keep the	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 28 and is configured such that the idle time period is one: during which, no packet is sent or received by the apparatus in the non-TCP connection (e.g., QUIC connection, etc.) to keep the non-TCP connection active.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However,</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	non-TCP connection active.	<p><u>state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>
39	The apparatus of claim 1, wherein the apparatus is configured such that, during the at least portion of the first non-TCP connection including the at least portion of the first non-TCP connection set up, the detection of the first time period of the first type of time period is based on the first duration, by being only based on whether another non-TCP packet is not received for the first non-TCP connection set up for the first duration.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that, during the at least portion of the first non-TCP connection (e.g., QUIC connection, etc.) including the at least portion of the first non-TCP connection set up, the detection of the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), by being only based on whether another non-TCP packet is not received for the first non-TCP connection set up for the first duration.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p><u>“A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
48	The apparatus of claim 1, wherein the apparatus is configured such that the first type of time period is started in response to at least one of a received packet or a sent packet.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that the first type of time period (e.g., idle-type timeout, etc.) is started in response to at least one of a received packet or a sent packet.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
51	The apparatus of claim 48, wherein the apparatus is configured such that, after the first duration is changed to the second duration, the detection of the second time period of the first type of time period is based on the	Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 48 and is configured such that, after the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is changed to the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), the detection of the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the second duration, by being only based on whether, for the second duration after the second time period of the first type of time period is started, no non-empty packet is received (e.g.,

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	second duration, by being only based on whether, for the second duration after the second time period of the first type of time period is started, no non-empty packet is received and processed to keep the first non-TCP connection at least partially active.	<p>connection remains idle, etc.) and processed to keep the first non-TCP connection (e.g., QUIC connection, etc.) at least partially active.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p><u>“A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</u></p> <p><u>QUIC: A UDP-Based Multiplexed and Secure Transport</u> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</u></p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><u>QUIC: A UDP-Based Multiplexed and Secure Transport</u> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2)

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
52	The apparatus of claim 51, wherein the apparatus is configured such that, during the second time period of the first type of time period, an empty packet is sent without causing a timer utilized to detect the second time period to be reset.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 51 and is configured such that, during the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.), an empty packet is sent (i.e. a packet carries no application data, etc.) without causing a timer utilized to detect the second time period to be reset.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, keep-alive signals are not required as a result of using idle timers and reducing the number of keep-alive signals to zero and, without a negotiated idle timeout, neither endpoint of a connection knows when the other side might timeout, so keep-alives (e.g., PINGs) have to be sent at unnecessarily shorter intervals to</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>insure the connection to stay active. Further, with a negotiated idle time period, only one side sends one or more keep-alives, if a keep-alive is used.</p> <p>The QUIC IETF transport spec recognizes the need to avoid unnecessary keep-alives (e.g., PINGs). See section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2 and section 5.1 of the IETF HTTP/3 draft at https://tools.ietf.org/id/draft-ietf-quic-http-34.html#name-idle-connections.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to <u>verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</u></p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p>
55	The apparatus of claim 1, wherein the apparatus is configured such that, after the first duration is changed to the second duration, the detection of the second time period of the first type of time period is based on the	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that, after the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is changed to the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), the detection of the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the second duration, by being only based on whether, for the second duration, no non-empty packet is received (e.g., connection remains idle, etc.) and processed to keep the first non-TCP connection (e.g., QUIC connection, etc.) at least partially active.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	second duration, by being only based on whether, for the second duration, no non-empty packet is received and processed to keep the first non-TCP connection at least partially active.	<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p><u>“A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</u></p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3)

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
56	The apparatus of claim 55, wherein the apparatus is configured such that, during the second time period of the first type of time period, an empty packet is sent without causing a timer utilized to detect the second time period to be reset.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 55 and is configured such that, during the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.), an empty packet is sent (i.e. a packet carries no application data, etc.) without causing a timer utilized to detect the second time period to be reset.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, keep-alive signals are not required as a result of using idle timers and reducing the number of keep-alive signals to zero and, without a negotiated idle timeout, neither endpoint of a connection knows when the other side might timeout, so keep-alives (e.g., PINGs) have to be sent at unnecessarily shorter intervals to</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>insure the connection to stay active. Further, with a negotiated idle time period, only one side sends one or more keep-alives, if a keep-alive is used.</p> <p>The QUIC IETF transport spec recognizes the need to avoid unnecessary keep-alives (e.g., PINGs). See section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2 and section 5.1 of the IETF HTTP/3 draft at https://tools.ietf.org/id/draft-ietf-quic-http-34.html#name-idle-connections.</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to <u>verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</u></p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p>
63	The apparatus of claim 1, wherein the apparatus is configured such that, during the at least portion of the first non-TCP connection including the at least portion of the first non-TCP connection set up, the detection of the	Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that, during the at least portion of the first non-TCP connection (e.g., QUIC connection, etc.) including the at least portion of the first non-TCP connection set up, the detection of the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), by being based on whether, for the first duration, no packet is received (e.g., connection remains idle, etc.) and processed to keep the first non-TCP connection at least partially active.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	first time period of the first type of time period is based on the first duration, by being based on whether, for the first duration, no packet is received and processed to keep the first non-TCP connection at least partially active.	<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: <code>original_connection_id</code>, <code>preferred_address</code>, <code>stateless_reset_token</code>, <code>ack_delay_exponent</code> and <code>active_connection_id_limit</code>. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • <code>initial_max_data</code> • <code>initial_max_stream_data_bidi_local</code> • <code>initial_max_stream_data_bidi_remote</code> • <code>initial_max_stream_data_uni</code> • <code>initial_max_streams_bidi</code> • <code>initial_max_streams_uni</code> <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes <code>initial_max_data</code> and either <code>initial_max_streams_bidi</code> and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
64	The apparatus of claim 1, wherein the apparatus is configured such that the first duration is not negotiated during the first non-TCP connection set up between the first and second nodes for the first non-TCP connection, and the second duration is negotiated between the first and second nodes for the first non-TCP connection.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured such that the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is not negotiated during the first non-TCP connection (e.g., QUIC connection, etc.) set up between the first and second nodes for the first non-TCP connection, and the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is negotiated between the first and second nodes for the first non-TCP connection.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a <u>QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient</u> of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3)

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
65	The apparatus of claim 1, wherein the apparatus is configured to, in response to detecting, by the first node, the third time period that is a second type of time period including an acknowledgment timeout period during which no acknowledgement packet is received in the first non-TCP connection and processed as an acknowledgement for a sent packet sent in the first non-TCP connection:	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured wherein the apparatus is configured to, in response to detecting, by the first node (e.g., one of a QUIC-compliant server or client, etc.), the third time period (e.g., probe timeout, etc.) that is a second type of time period (e.g., probe-type timeout, etc.) including an acknowledgment timeout period during which no acknowledgement packet is received in the first non-TCP connection (e.g., QUIC connection, etc.) and processed as an acknowledgement for a sent packet sent in the first non-TCP connection:</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions ...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	unilaterally at least partially close the first	Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured to unilaterally at least partially close (e.g.,

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	non-TCP connection, by releasing, by the first node, the first resource allocated for the first non-TCP connection,	<p>terminate, etc.) the first non-TCP connection (e.g., QUIC connection, etc.), by releasing, by the first node (e.g., one of a QUIC-compliant server or client, etc.), the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) allocated for the first non-TCP connection,</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <u>idle_timeout</u> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily detected based on the value received in the <u>idle_timeout</u> field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC <u>connection can be terminated</u> in one of three ways:</p> <ul style="list-style-type: none"> • <u>idle timeout</u> (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p>
	wherein: the first non-TCP connection set up further includes, in addition to (a-	Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured wherein: the first non-TCP connection (e.g., QUIC connection, etc.) set up further includes, in addition to (a-1) sending the first

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	1) sending the first non-TCP packet that is also for use in determining the third duration of the third time period:	<p>non-TCP packet (e.g., first QUIC negotiation packet, etc.) that is also for use in determining the third duration (e.g., duration associated with probe timeout, etc.) of the third time period (e.g., probe timeout, etc.):</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>$PTO = \text{smoothed_rtt} + \max(4 * \text{rttvar}, k\text{Granularity}) + \text{max_ack_delay}$</p> <p>$k\text{Granularity}$, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate ($4 * \text{rttvar}$), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least $k\text{Granularity}$, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>(a-2) after sending the first non-TCP packet and without any other communication between the first node and the second node after (a-1) and before (a-2), receiving, at the first node from the second node, a second non-TCP packet, and (a-3) after receiving the second non-TCP packet and without any other communication</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured wherein (a-2) after sending the first non-TCP packet (e.g., first QUIC negotiation packet, etc.) and without any other communication between the first node (e.g., one of a QUIC-compliant server or client, etc.) and the second node (e.g., the other one of the QUIC-compliant server or client, etc.) after (a-1) and before (a-2), receiving, at the first node from the second node, a second non-TCP packet (e.g., second QUIC negotiation packet, etc.), and (a-3) after receiving the second non-TCP packet and without any other communication between the first node and the second node after (a-2) and before (a-3), sending, from the first node to the second node, a third non-TCP packet, and further wherein:</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	between the first node and the second node after (a-2) and before (a-3), sending, from the first node to the second node, a third non-TCP packet, and further wherein:	<p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																
	the first duration, the second duration, and the	Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured wherein the first duration (e.g., duration																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	third duration are capable of being of different durations;	<p>associated with idle timeout in connection with 0-RTT packets, etc.), the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), and the third duration (e.g., duration associated with probe timeout, etc.) are capable of being of different durations;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“<u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the second time period and the third time period are capable of being detected during the first non-TCP connection set up and	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured wherein the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) and the third time period (e.g., probe timeout, etc.) are capable of being detected during the first non-TCP connection (e.g., QUIC connection, etc.) set up and thereafter during the first non-TCP connection; and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	thereafter during the first non-TCP connection; and	<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> <p>$\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time” <i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>the first time period and the second time period are started in response to at least one of a received packet or a sent packet; and</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured wherein the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) and the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) are started in response to at least one of a received packet or a sent packet; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> <p>PTO = smoothed_rtt + max(4*rttvar, kGranularity) + max_ack_delay</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time” <i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>during the at least portion of the first non-TCP connection including the at least portion of the first non-TCP connection set up, the detection of the first time period of the first type of time period is based on the first duration, by being based on whether, during the first duration, no packet is received and processed to keep the first non-TCP connection at least partially active; and</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured wherein, during the at least portion of the first non-TCP connection (e.g., QUIC connection, etc.) including the at least portion of the first non-TCP connection set up, the detection of the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the first duration, by being based on whether, during the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), no packet is received (e.g., connection remains idle, etc.) and processed to keep the first non-TCP connection at least partially active; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets. Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
	after the first duration is changed to the second duration, the detection of the second time period of the first type of time period is based on the second duration, by being based on whether, during the second duration, no packet is received and processed to keep the first non-TCP connection at least partially active;	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured wherein, after the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is changed to the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), the detection of the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the second duration, by being based on whether, during the second duration, no packet is received (e.g., connection remains idle, etc.) and processed to keep the first non-TCP connection (e.g., QUIC connection, etc.) at least partially active;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p><u>“A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	where only one of the first time period, the second time period, or the third time period is detected for the first non-TCP connection.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 1 and is configured where only one of the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.), the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.), or the third time period (e.g., probe timeout, etc.) is detected for the first non-TCP connection (e.g., QUIC connection, etc.).</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
67	The apparatus of claim 65, wherein the apparatus is configured such that the first type of time period is started in response to the received packet.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured such that the first type of time period (e.g., idle-type timeout, etc.) is started in response to the received packet.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
68	The apparatus of claim 65, wherein the apparatus is configured such that the first type of time period is started in response to the sent packet.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured such that the first type of time period (e.g., idle-type timeout, etc.) is started in response to the sent packet.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																
69	The apparatus of claim 65, wherein the apparatus is	Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured such that, during the second time period																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	configured such that, during the second time period of the first type of time period, an empty packet is sent without causing a timer utilized to detect the second time period to be reset.	<p>(e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.), an empty packet is sent (i.e. a packet carries no application data, etc.) without causing a timer utilized to detect the second time period to be reset.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: Based upon information and belief, keep-alive signals are not required as a result of using idle timers and reducing the number of keep-alive signals to zero and, without a negotiated idle timeout, neither endpoint of a connection knows when the other side might timeout, so keep-alives (e.g., PINGs) have to be sent at unnecessarily shorter intervals to insure the connection to stay active. Further, with a negotiated idle time period, only one side sends one or more keep-alives, if a keep-alive is used.</p> <p>The QUIC IETF transport spec recognizes the need to avoid unnecessary keep-alives (e.g., PINGs). See section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2 and section 5.1 of the IETF HTTP/3 draft at https://tools.ietf.org/id/draft-ietf-quic-http-34.html#name-idle-connections.</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to <u>verify that their peers are still alive or to check reachability to the peer.</u> The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
71	The apparatus of claim 65, wherein the apparatus is configured such that the third duration is determined using a second parameter field that is communicated in the first non-TCP connection and that identifies second metadata.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured such that the third duration (e.g., duration associated with probe timeout, etc.) is determined using a second parameter field (e.g., second idle_timeout parameter field, etc.) that is communicated in the first non-TCP connection (e.g., QUIC connection, etc.) and that identifies second metadata (e.g., second idle timeout value, etc.).</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
72	The apparatus of claim 71, wherein the apparatus is configured such that the second duration is determined using the first parameter field identifying the first metadata, and the second parameter field identifying the second metadata.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 71 and is configured such that the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is determined using the first parameter field (e.g., first idle_timeout parameter field, etc.) identifying the first metadata (e.g., first idle timeout value, etc.), and the second parameter field (e.g., second idle_timeout parameter field, etc.) identifying the second metadata (e.g., second idle timeout value, etc.).</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a <u>QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient</u> of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) <u>and three times the current Probe Timeout (PTO)</u>.</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>modified based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
73	The apparatus of claim 65, wherein the apparatus is configured such that: the first non-TCP packet is not a synchronize (SYN) packet;	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured such that: the first non-TCP packet (e.g., first QUIC negotiation packet, etc.) is not a synchronize (SYN) packet;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: The following serves to distinguish the use of SYN handshake packets from QUIC negotiation packets where parameter values like idle_timeout may be exchanged.</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <p>1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y</p> <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. <u>The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN.</u> The three way handshake and the advantages of a clock-driven scheme are discussed in [3].”</p> <p>“Request for Comments” (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
	the first non-TCP connection is not a TCP extension connection that involves a TCP extension;	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured wherein the first non-TCP connection (e.g., QUIC connection, etc.) is not a TCP extension connection that involves a TCP extension;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since <u>QUIC is built on top of UDP</u>, it suffers from no such limitations.”</p> <p>https://www.chromium.org/quic</p>
	<p>the first type of time period includes an idle time period that does not include a user timeout period;</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured wherein the first type of time period (e.g., idle-type timeout, etc.) includes an idle time period that does not include a user timeout period;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first time period is only capable of being detected during the first non-TCP connection set up;	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured wherein the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) is only capable of being detected during the first non-TCP connection (e.g., QUIC connection, etc.) set up;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>the first duration is not negotiated between the first and second nodes for the first non-TCP connection using (a-1), (a-2), nor (a-3), and the second duration and the third duration are negotiated between the first and second nodes for the first non-TCP connection using at least</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured wherein the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is not negotiated between the first and second nodes for the first non-TCP connection (e.g., QUIC connection, etc.) using (a-1), (a-2), nor (a-3), and the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) and the third duration (e.g., duration associated with probe timeout, etc.) are negotiated between the first and second nodes for the first non-TCP connection using at least one of (a-1), (a-2), or (a-3);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a <u>QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient</u> of such packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	one of (a-1), (a-2), or (a-3);	<p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, k\text{Granularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>the first duration is used and the second duration is not used, for detecting the first type of time period until the first duration is changed to the second duration;</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured wherein the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is used and the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is not used, for detecting the first type of time period (e.g., idle-type timeout, etc.) until the first duration is changed to the second duration;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</u></p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>the second duration is determined based on a first algorithm, and the third duration is determined based on a second algorithm that is different from the first algorithm;</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured wherein the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is determined based on a first algorithm (e.g., used to negotiate an idle timeout period between nodes, etc.), and the third duration (e.g., duration associated with probe timeout, etc.) is determined based on a second algorithm (e.g., used to determine the probe timeout, etc.) that is different from the first algorithm;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a <u>QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient</u> of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quick-transport-22</p> <p>Note: Based upon information and belief, keep-alive signals are not required as a result of using idle timers and reducing the number of keep-alive signals to zero and, without a negotiated idle timeout, neither endpoint of a connection knows when the other side might timeout, so keep-alives (e.g., PINGs) have to be sent at unnecessarily shorter intervals to insure the connection to stay active. Further, with a negotiated idle time period, only one side sends one or more keep-alives, if a keep-alive is used.</p> <p>The QUIC IETF transport spec recognizes the need to avoid unnecessary keep-alives (e.g., PINGs). See section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quick-transport-34#section-10.1.2 and section 5.1 of the IETF HTTP/3 draft at https://tools.ietf.org/id/draft-ietf-quick-http-34.html#name-idle-connections.</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p>
	the third time period is detected only in response to no acknowledgement packet being received in the first non-TCP connection and processed as an acknowledgement, during the third duration, for a sent packet sent in the first non-TCP connection;	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured wherein the third time period (e.g., probe timeout, etc.) is detected only in response to no acknowledgement packet being received in the first non-TCP connection (e.g., QUIC connection, etc.) and processed as an acknowledgement, during the third duration (e.g., duration associated with probe timeout, etc.), for a sent packet sent in the first non-TCP connection;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>during the at least portion of the first non-TCP connection including the at least portion of the first non-TCP connection set up, the detection of the first time period of the first type of time period is based on the first duration, by being only based on whether any non-TCP packet is received for the first non-TCP connection set up during the first duration;</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured wherein, during the at least portion of the first non-TCP connection (e.g., QUIC connection, etc.) including the at least portion of the first non-TCP connection set up, the detection of the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), by being only based on whether any non-TCP packet is received for the first non-TCP connection set up during the first duration;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	<p>after the first duration is changed to the second duration, the detection of the second time period of the first type of time period is based on the second duration, by being only based on whether, during the second duration, no packet is received and processed to keep the first non-TCP connection at least partially active, nor is expected to be received in the first non-TCP connection; and</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured wherein, after the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is changed to the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), the detection of the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the second duration, by being only based on whether, during the second duration, no packet is received (e.g., connection remains idle, etc.) and processed to keep the first non-TCP connection (e.g., QUIC connection, etc.) at least partially active, nor is expected to be received in the first non-TCP connection; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4)

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p><u>“A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787]</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</u> <u>QUIC: A UDP-Based Multiplexed and Secure Transport</u> https://tools.ietf.org/id/draft-ietf-quic-transport-27.txt</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>an idle timeout attribute is utilized for the first type of time period, and a user timeout attribute, that is separate from the idle timeout attribute, is utilized for controlling communication of one or more acknowledgment packets.</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 65 and is configured wherein an idle timeout attribute is utilized for the first type of time period (e.g., idle-type timeout, etc.), and a user timeout attribute (e.g., acknowledgement in the form of PING frames, etc.), that is separate from the idle timeout attribute, is utilized for controlling communication of one or more acknowledgment packets.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“19.2. PING Frame</p> <p><u>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer.</u> The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2
74	The apparatus of claim 73, wherein the apparatus is configured such that: the third time period is detected utilizing a timer that is not utilized to detect either of the first and second time periods; and	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 73 and is configured such that: the third time period (e.g., probe timeout, etc.) is detected utilizing a timer that is not utilized to detect either of the first and second time periods; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quick-transport-22</p>
	the second duration is the same as the first duration, despite being capable of being different from the first duration.	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) that infringes claim 73 and is configured wherein the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is the same as the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), despite being capable of being different from the first duration.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: <code>original_connection_id</code>, <code>preferred_address</code>, <code>stateless_reset_token</code>, <code>ack_delay_exponent</code> and <code>active_connection_id_limit</code>. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • <code>initial_max_data</code> • <code>initial_max_stream_data_bidi_local</code> • <code>initial_max_stream_data_bidi_remote</code> • <code>initial_max_stream_data_uni</code> • <code>initial_max_streams_bidi</code> • <code>initial_max_streams_uni</code> <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes <code>initial_max_data</code> and either <code>initial_max_streams_bidi</code> and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
75	A non-transitory computer readable storage medium storing one or more programs, the one or more programs comprising instructions, which when executed by a first node, cause the first node to:	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium storing one or more programs (e.g. HTML pages, etc.), the one or more programs comprising instructions (e.g., server software, etc.), which when executed by a first node (e.g., one of a QUIC-compliant server or client, etc.), cause the first node to:</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>we're happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers."</p>

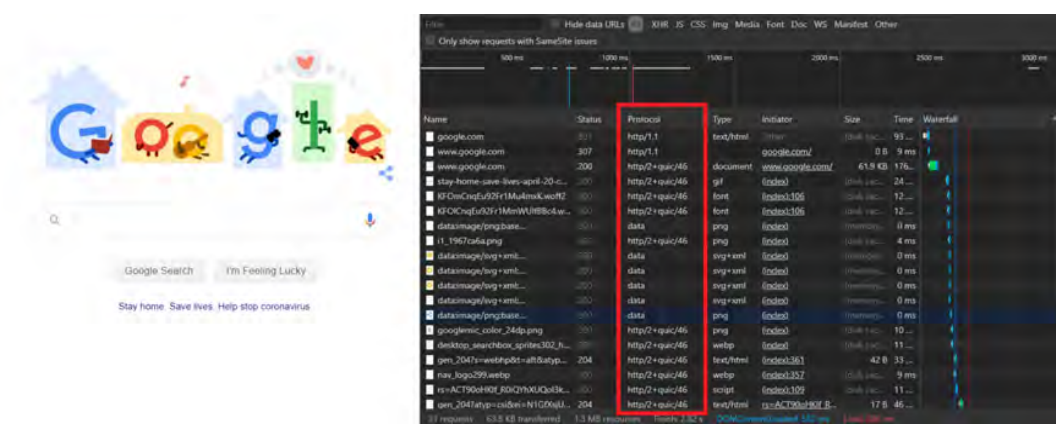
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p> <p> https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 </p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p> https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27 </p> <p> Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities. </p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Note: Below is a web page of Google (https://www.google.com/).</p> 
	receive first information on which at least a first duration for detecting a first type of time period is based;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium is configured to receive first information (e.g., first value, etc.) on which at least a first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) for detecting a first type of time period (e.g., idle-type timeout, etc.) is based;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the idle timeout field of the connection negotiation packet.</u></p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“3.3. Session resumption versus Keep-alive</p> <p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p> <ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods. • <u>Resume the session after a long idle period, using 0-RTT resumption when appropriate.</u> <p>...</p> <p>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic. In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. <u>When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with restarting the connection.</u> This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.”</p> <p>https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
	identify a first resource for a first connection, where the first connection does not utilize Transmission Control Protocol (TCP);	Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium is configured to identify a first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) for a first connection (e.g., QUIC

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>connection, etc.), where the first connection does not utilize Transmission Control Protocol (TCP);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a <u>suite of cloud computing services</u> that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube” https://en.wikipedia.org/wiki/Google_Cloud_Platform</p>
	<p>generate a first packet including a first parameter field identifying first metadata for use in determining a second duration for detecting the first type of time period;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium is configured to generate a first packet (e.g., QUIC negotiation packet, etc.) including a first parameter field (e.g., first idle_timeout parameter field, etc.) identifying first metadata (e.g., first idle timeout value, etc.) for use in determining a second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) for detecting the first type of time period (e.g., idle-type timeout, etc.);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.” <i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	set up the first connection, by sending, from the first node to a second node, the first packet to provide the first metadata to the second node, for use by	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium is configured to set up the first connection (e.g., QUIC connection, etc.), by sending, from the first node (e.g., one of a QUIC-compliant server or client, etc.) to a second node (e.g., the other one of the QUIC-compliant server or client, etc.), the first packet (e.g., QUIC negotiation packet, etc.) to provide the first metadata (e.g., first idle timeout value, etc.) to the</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	the second node in determining the second duration for detecting the first type of time period;	<p>second node, for use by the second node in determining the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) for detecting the first type of time period (e.g., idle-type timeout, etc.);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	in response to detecting, based on the first duration and by the first node during at least a portion of the first connection including at least a portion of the first connection set up, a first time period of the first type of time period, at least partially close the first connection, by releasing, by the first	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium is configured to, in response to detecting, based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) and by the first node (e.g., one of a QUIC-compliant server or client, etc.) during at least a portion of the first connection (e.g., QUIC connection, etc.) including at least a portion of the first connection set up, a first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.), at least partially close (e.g., terminate, etc.) the first connection, by releasing, by the first node, the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) for the first connection;</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	node, the first resource for the first connection;	<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <u>idle_timeout</u> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily detected based on the value received in the <u>idle_timeout</u> field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established <u>QUIC connection</u> can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • <u>idle timeout</u> (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	in response to detecting, based on the second duration and by the first node after the first	Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium is configured to, in response to detecting, based on the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) and by the first node

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	duration is changed to the second duration, a second time period of the first type of time period, at least partially close the first connection, by releasing, by the first node, the first resource for the first connection; and	<p>(e.g., one of a QUIC-compliant server or client, etc.) after the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is changed to the second duration, a second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.), at least partially close (e.g., terminate, etc.) the first connection (e.g., QUIC connection, etc.), by releasing, by the first node, the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) for the first connection; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>packets</u>. Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: <code>original_connection_id</code>, <code>preferred_address</code>, <code>stateless_reset_token</code>, <code>ack_delay_exponent</code> and <code>active_connection_id_limit</code>. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • <code>initial_max_data</code> • <code>initial_max_stream_data_bidi_local</code> • <code>initial_max_stream_data_bidi_remote</code> • <code>initial_max_stream_data_uni</code> • <code>initial_max_streams_bidi</code> • <code>initial_max_streams_uni</code>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	in response to detecting, based on a third duration and by the first node, a third time period, at least partially close the first connection, by releasing, by the first node, the first resource allocated for the first connection, the third duration being determined	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium is configured to, in response to detecting, based on a third duration (e.g., duration associated with probe timeout, etc.) and by the first node (e.g., one of a QUIC-compliant server or client, etc.), a third time period (e.g., probe timeout, etc.), at least partially close (e.g., terminate, etc.) the first connection (e.g., QUIC connection, etc.), by releasing, by the first node, the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) allocated for the first connection, the third duration being determined based on a first algorithm (e.g., used to determine the probe timeout, etc.) that is different from a second algorithm (e.g., used to negotiate an idle timeout period</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	based on a first algorithm that is different from a second algorithm on which a determination of the second duration is based.	<p>between nodes, etc.) on which a determination of the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is based.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt),</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>the variance in the estimate ($4 \cdot \text{rttvar}$), and <code>max_ack_delay</code>, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least <code>kGranularity</code>, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p>
76	The non-transitory computer readable storage medium of claim 75, wherein the instructions, when executed by the first node, cause the first node to: in response to detecting, by the first node, the third time period that is a second type of time period including an acknowledgment timeout period during which no acknowledgement packet	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 75 and is configured wherein the instructions (e.g., server software, etc.), when executed by the first node (e.g., one of a QUIC-compliant server or client, etc.), cause the first node to: in response to detecting, by the first node, the third time period (e.g., probe timeout, etc.) that is a second type of time period (e.g., probe-type timeout, etc.) including an acknowledgment timeout period during which no acknowledgement packet is received in the first connection (e.g., QUIC connection, etc.) and processed as an acknowledgement for a sent packet sent in the first connection:</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions ...</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	is received in the first connection and processed as an acknowledgement for a sent packet sent in the first connection:	<ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt),</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>the variance in the estimate ($4 \cdot \text{rttvar}$), and <code>max_ack_delay</code>, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least <code>kGranularity</code>, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	release, by the first node, the first resource for the first connection, wherein:	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 75 and is configured to release, by the first node (e.g., one of a QUIC-compliant server or client, etc.), the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/rcv buffers, etc.) for the first connection (e.g., QUIC connection, etc.), wherein:</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“10. <u>Connection Termination</u></p> <p>An <u>established QUIC connection can be terminated</u> in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY <u>discard connection state</u> if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>If the idle timeout is enabled, a <u>connection is silently closed and the state is discarded</u> when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>the first time period and the second time period are started in response to at least one of a received packet or a sent packet; and</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 75 and is configured wherein the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) and the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) are started in response to at least one of a received packet or a sent packet; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt),</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>the variance in the estimate ($4 \cdot \text{rttvar}$), and <code>max_ack_delay</code>, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least <code>kGranularity</code>, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>during the at least portion of the first connection including the at least portion of the first connection set up, the detection of the first time period of the first type of time period is based on the first duration, by being based on whether, during the first duration after the first time period is started, no packet is received and processed to keep the first connection at least partially active; and</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 75 and is configured wherein, during the at least portion of the first connection (e.g., QUIC connection, etc.) including the at least portion of the first connection set up, the detection of the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), by being based on whether, during the first duration after the first time period is started, no packet is received (e.g., connection remains idle, etc.) and processed to keep the first connection at least partially active; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quick-transport-22</p>
	based on the first duration being changed to the second duration, the detection of the second time period of the first type of time period is based on the second duration, by being based on whether, during the second duration after the	Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 75 and is configured wherein, based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) being changed to the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), the detection of the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the second duration, by being based on whether, during the second duration after the second time period is started, no packet is received (e.g., connection remains idle, etc.) and processed to keep the first connection (e.g., QUIC connection, etc.) at least partially active.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	second time period is started, no packet is received and processed to keep the first connection at least partially active.	<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quick-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys,

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
77	The non-transitory computer readable storage medium of claim 76, wherein the first connection set up further includes, in addition to (a-1) sending the first packet that is also for use in determining the third duration of the third time period:	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein the first connection (e.g., QUIC connection, etc.) set up further includes, in addition to (a-1) sending the first packet that is also for use in determining the third duration (e.g., duration associated with probe timeout, etc.) of the third time period (e.g., probe timeout, etc.):</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions ...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	<p>(a-2) after sending the first packet and without any other communication between the first node and the second node after (a-1) and before (a-2), receiving, at the first node from the second node, a second packet, and (a-3) after receiving the second packet and without any other communication between the first node and the second node after (a-2) and before (a-3), sending, from the first node to the second node, a third packet, and further wherein:</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein, (a-2) after sending the first packet and without any other communication between the first node (e.g., one of a QUIC-compliant server or client, etc.) and the second node (e.g., the other one of the QUIC-compliant server or client, etc.) after (a-1) and before (a-2), receiving, at the first node from the second node, a second packet, and (a-3) after receiving the second packet and without any other communication between the first node and the second node after (a-2) and before (a-3), sending, from the first node to the second node, a third packet, and further wherein:</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																
	the first duration, the second duration, and the	Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	third duration are capable of being of different durations;	<p>that infringes claim 76 and is configured wherein the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), and the third duration (e.g., duration associated with probe timeout, etc.) are capable of being of different durations;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“<u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)”

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
	the second time period and the third time period are capable of being detected during the first connection set up and thereafter during the first connection;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) and the third time period (e.g., probe timeout, etc.) are capable of being detected during the first connection (e.g., QUIC connection, etc.) set up and thereafter during the first connection;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions ... • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, k\text{Granularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first packet is not a synchronize (SYN) packet;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein the first packet is not a synchronize (SYN) packet;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: The following serves to distinguish the use of SYN handshake packets from QUIC negotiation packets where parameter values like idle_timeout may be exchanged.</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <p>1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y</p> <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. <u>The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN.</u> The three way handshake and the advantages of a clock-driven scheme are discussed in [3].”</p> <p>“Request for Comments” (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
	the first connection does not utilize a TCP extension;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein the first connection (e.g., QUIC connection, etc.) does not utilize a TCP extension;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since <u>QUIC is built on top of UDP</u>, it suffers from no such limitations.”</p> <p>https://www.chromium.org/quic</p>
	<p>the first type of time period includes an idle time period that does not include a user timeout period;</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein the first type of time period (e.g., idle-type timeout, etc.) includes an idle time period that does not include a user timeout period;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first time period is only capable of being detected during the first connection set up;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) is only capable of being detected during the first connection (e.g., QUIC connection, etc.) set up;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>packets</u>. Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: <code>original_connection_id</code>, <code>preferred_address</code>, <code>stateless_reset_token</code>, <code>ack_delay_exponent</code> and <code>active_connection_id_limit</code>. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • <code>initial_max_data</code> • <code>initial_max_stream_data_bidi_local</code> • <code>initial_max_stream_data_bidi_remote</code> • <code>initial_max_stream_data_uni</code> • <code>initial_max_streams_bidi</code> • <code>initial_max_streams_uni</code>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first duration is not negotiated between the first and second nodes for the first connection using (a-1), (a-2), nor (a-3), and the second duration and the third duration are negotiated between the first and second nodes for the first connection using	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is not negotiated between the first and second nodes for the first connection (e.g., QUIC connection, etc.) using (a-1), (a-2), nor (a-3), and the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) and the third duration (e.g., duration associated with probe timeout, etc.) are negotiated between the first and second nodes for the first connection using at least one of (a-1), (a-2), or (a-3);</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	at least one of (a-1), (a-2), or (a-3);	<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a <u>QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient</u> of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first duration is used and the second duration is not used, for detecting the first type of time period until the first duration is	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is used and the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is not</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	changed to the second duration;	<p>used, for detecting the first type of time period (e.g., idle-type timeout, etc.) until the first duration is changed to the second duration;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</u></p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the second duration is determined based on a first algorithm, and the third duration is determined based on a second algorithm that is different from the first algorithm;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is determined based on a first algorithm (e.g., used to negotiate an idle timeout period between nodes, etc.), and the third duration (e.g., duration associated with probe timeout, etc.) is determined based on a second algorithm (e.g., used to determine the probe timeout, etc.) that is different from the first algorithm;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a <u>QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient</u> of such packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quick-transport-22</p> <p>Note: Based upon information and belief, keep-alive signals are not required as a result of using idle timers and reducing the number of keep-alive signals to zero and, without a negotiated idle timeout, neither endpoint of a connection knows when the other side might timeout, so keep-alives (e.g., PINGs) have to be sent at unnecessarily shorter intervals to insure the connection to stay active. Further, with a negotiated idle time period, only one side sends one or more keep-alives, if a keep-alive is used.</p> <p>The QUIC IETF transport spec recognizes the need to avoid unnecessary keep-alives (e.g., PINGs). See section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quick-transport-34#section-10.1.2 and section</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>5.1 of the IETF HTTP/3 draft at https://tools.ietf.org/id/draft-ietf-quic-http-34.html#name-idle-connections.</p> <p>“19.2. PING Frame</p> <p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p>
	the third time period is detected only in response to no acknowledgement packet being received in the first connection and processed as an	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein the third time period (e.g., probe timeout, etc.) is detected only in response to no acknowledgement packet being received in the first connection (e.g., QUIC connection, etc.) and processed as an acknowledgement, during the</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	acknowledgement, during the third duration, for a sent packet sent in the first connection;	<p>third duration (e.g., duration associated with probe timeout, etc.), for a sent packet sent in the first connection;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>$PTO = smoothed_rtt + \max(4 * rttvar, kGranularity) + max_ack_delay$</p> <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	during the at least portion of the first connection including the at least portion of the first connection set up, the detection of the first time period of the first type of time period is based on the first duration, by being only based on whether any packet is received for the first connection set up during the first duration;	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein, during the at least portion of the first connection (e.g., QUIC connection, etc.) including the at least portion of the first connection set up, the detection of the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), by being only based on whether any packet is received for the first connection set up during the first duration;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>based on the first duration being changed to the second duration, the detection of the second time period of the first type of time period is based on the second duration, by being only based on whether, during the second duration, no packet is received and processed to keep the first connection at least partially active, nor is expected to be received in the first connection; and</p>	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein, based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) being changed to the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), the detection of the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the second duration, by being only based on whether, during the second duration, no packet is received (e.g., connection remains idle, etc.) and processed to keep the first connection (e.g., QUIC connection, etc.) at least partially active, nor is expected to be received in the first connection; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	an idle timeout attribute is utilized for the first type of time period, and a user timeout attribute, that is separate from the idle timeout attribute, is	<p>Google, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform with a non-transitory computer readable storage medium that infringes claim 76 and is configured wherein an idle timeout attribute is utilized for the first type of time period (e.g., idle-type timeout, etc.), and a user timeout attribute (e.g., acknowledgement in the form of PING frames, etc.), that is separate from the idle timeout attribute, is utilized for controlling communication of one or more acknowledgment packets.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	utilized for controlling communication of one or more acknowledgment packets.	<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“19.2. PING Frame</p> <p><u>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer.</u> The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p>
78	A method, comprising: at at least a portion of a first node: identifying first information on which at least a first duration for detecting a first type of time period is based;	<p>Google owns, controls, and/or uses at least one server computer, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to that performs a method including, at at least a portion of a first node (e.g., one of a QUIC-compliant server or client, etc.): identifying first information (e.g., first value, etc.) on which at least a first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) for detecting a first type of time period (e.g., idle-type timeout, etc.) is based;</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“3.3. Session resumption versus Keep-alive</p> <p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p> <ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods. • <u>Resume the session after a long idle period, using 0-RTT resumption when appropriate.</u> <p>...</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic. In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. <u>When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with restarting the connection.</u> This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.”</p> <p>https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
	allocating a first resource for a first Transmission Control Protocol-variant (TCP-variant) connection;	<p>Google owns, controls, and/or uses at least one server computer, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform for allocating a first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) for a first Transmission Control Protocol-variant (TCP-variant) connection (e.g., QUIC connection, etc.);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a <u>suite of cloud computing services</u> that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube”</p> <p>https://en.wikipedia.org/wiki/Google_Cloud_Platform</p>
	generating a first TCP-variant packet including a first parameter field identifying first metadata for use in determining a second duration for detecting the first type of time period;	<p>Google owns, controls, and/or uses at least one server computer, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform for generating a first TCP-variant packet including a first parameter field (e.g., first idle_timeout parameter field, etc.) identifying first metadata (e.g., first idle timeout value, etc.) for use in determining a second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) for detecting the first type of time period (e.g., idle-type timeout, etc.);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	setting up the first TCP-variant connection, by sending, from the first node to a second node, the first TCP-variant packet to provide the first metadata	<p>Google owns, controls, and/or uses at least one server computer, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform for setting up the first TCP-variant connection (e.g., QUIC connection, etc.), by sending, from the first node (e.g., one of a QUIC-compliant server or client, etc.) to a second node (e.g., the other one of the QUIC-compliant server or client, etc.), the first TCP-variant packet to provide the first metadata (e.g., first idle timeout value, etc.) to the second node, for use by</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	to the second node, for use by the second node in determining the second duration for detecting the first type of time period;	<p>the second node in determining the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) for detecting the first type of time period (e.g., idle-type timeout, etc.);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	based on detecting, by the first node and during at least a portion of the first TCP-variant connection including at least a portion of the first TCP-variant connection set up, a first duration-based first time period of the first type of time period, at least partially closing the first TCP-variant connection,	<p>Google owns, controls, and/or uses at least one server computer, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform for, based on detecting, by the first node (e.g., one of a QUIC-compliant server or client, etc.) and during at least a portion of the first TCP-variant connection (e.g., QUIC connection, etc.) including at least a portion of the first TCP-variant connection set up, a first duration-based first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.), at least partially closing the first TCP-variant connection, by releasing the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) allocated for the first TCP-variant connection;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	by releasing the first resource allocated for the first TCP-variant connection;	<p>Note: As set forth below, since the <u>idle_timeout</u> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily detected based on the value received in the <u>idle_timeout</u> field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC <u>connection can be terminated</u> in one of three ways:</p> <ul style="list-style-type: none"> • <u>idle timeout</u> (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	based on detecting, by the first node and based on the first duration being changed to the second duration, a second	Google owns, controls, and/or uses at least one server computer, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform for, based on detecting, by the first node (e.g., one of a QUIC-compliant server or client, etc.) and based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) being changed to the second duration (e.g., duration associated with idle

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	duration-based second time period of the first type of time period, at least partially closing the first TCP-variant connection, by releasing the first resource allocated for the first TCP-variant connection; and	<p>timeout in connection with 1-RTT packets, etc.), a second duration-based second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.), at least partially closing the first TCP-variant connection (e.g., QUIC connection, etc.), by releasing the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) allocated for the first TCP-variant connection; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>based on detecting, by the first node and based on a third duration, a third duration-based third time period, at least partially closing the first TCP-variant connection, by releasing the first resource allocated for the first TCP-variant connection, where the third duration is determined based on a first algorithm that is different</p>	<p>Google owns, controls, and/or uses at least one server computer, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform for, based on detecting, by the first node (e.g., one of a QUIC-compliant server or client, etc.) and based on a third duration (e.g., duration associated with probe timeout, etc.), a third duration-based third time period (e.g., probe timeout, etc.), at least partially closing the first TCP-variant connection (e.g., QUIC connection, etc.), by releasing the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) allocated for the first TCP-variant connection, where the third duration is determined based on a first algorithm (e.g., used to determine the probe timeout, etc.) that is different from a second algorithm (e.g., used to negotiate an idle timeout period between nodes, etc.) on which a determination of the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is based.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	from a second algorithm on which a determination of the second duration is based.	<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p>
80	The method of claim 78, wherein the first time period is only capable of being detected during the first TCP-variant connection set up.	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) is only capable of being detected during the first TCP-variant connection (e.g., QUIC connection, etc.) set up.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
83	The method of claim 78, wherein the third time period is detected based on no acknowledgement packet being detected, for	Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the third time period (e.g., probe timeout, etc.) is detected based on no acknowledgement packet being detected, for the third duration (e.g., duration associated with probe timeout, etc.) that starts based on a sent packet being sent in the first TCP-variant connection (e.g., QUIC

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	the third duration that starts based on a sent packet being sent in the first TCP-variant connection, as being received in the first TCP-variant connection and processed to acknowledge the sent packet.	<p>connection, etc.), as being received in the first TCP-variant connection and processed to acknowledge the sent packet.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions ... • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>$PTO = smoothed_rtt + \max(4 * rttvar, kGranularity) + \max_ack_delay$</p> <p>$kGranularity$, $smoothed_rtt$, $rttvar$, and \max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time ($smoothed_rtt$), the variance in the estimate ($4 * rttvar$), and \max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least $kGranularity$, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
84	The method of claim 83, wherein the first TCP-variant packet also includes data for use in determining the third time period.	<p>Google infringes claim 83, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first TCP-variant packet (e.g., first QUIC negotiation packet, etc.) also includes data for use in determining the third time period (e.g., probe timeout, etc.).</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
86	<p>The method of claim 83, wherein the first duration is not negotiated between the first and second nodes during the first TCP-variant connection set up, the second duration is negotiated between the first and second nodes for the first TCP-variant connection, and the third duration of the third time period is negotiated between the first and second nodes for the first TCP-variant connection.</p>	<p>Google infringes claim 83, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is not negotiated between the first and second nodes during the first TCP-variant connection (e.g., QUIC connection, etc.) set up, the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is negotiated between the first and second nodes for the first TCP-variant connection, and the third duration (e.g., duration associated with probe timeout, etc.) of the third time period (e.g., probe timeout, etc.) is negotiated between the first and second nodes for the first TCP-variant connection.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
88	The method of claim 83, wherein the third time period is a second type of time period.	<p>Google infringes claim 83, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the third time period (e.g., probe timeout, etc.) is a second type of time period (e.g., probe-type timeout, etc.).</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
103	The method of claim 78, wherein the first type of time period includes an idle time period.	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first type of time period (e.g., idle-type timeout, etc.) includes an idle time period.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
104	The method of claim 103, wherein the idle time period is one: during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active.	<p>Google infringes claim 103, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the idle time period is one: during which, no packet is communicated in the TCP-variant connection (e.g., QUIC connection, etc.) to keep the TCP-variant connection active.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-27.txt</p>
159	The method of claim 78, and further comprising: based on detecting, by the first node, the third time period that is a second type of time period including an acknowledgment timeout period during which no	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method further comprising: based on detecting, by the first node (e.g., one of a QUIC-compliant server or client, etc.), the third time period (e.g., probe timeout, etc.) that is a second type of time period (e.g., probe-type timeout, etc.) including an acknowledgment timeout period during which no acknowledgement packet is received in the first TCP-variant connection (e.g., QUIC connection, etc.) and processed as an acknowledgement for a sent packet sent in the first TCP-variant connection:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	acknowledgement packet is received in the first TCP-variant connection and processed as an acknowledgement for a sent packet sent in the first TCP-variant connection:	<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions ... • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> <p>PTO = smoothed_rtt + max(4*rttvar, kGranularity) + max_ack_delay</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time” <i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. <u>The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint.</u> An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	unilaterally at least partially closing the first TCP-variant connection, by releasing, by the first node, the first resource allocated for the first TCP-variant connection, wherein:	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method for unilaterally at least partially closing the first TCP-variant connection (e.g., QUIC connection, etc.), by releasing, by the first node (e.g., one of a QUIC-compliant server or client, etc.), the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) allocated for the first TCP-variant connection, wherein:</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <u>idle timeout value sets the duration of idleness, after which the connection is shutdown</u>, a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC <u>connection can be terminated</u> in one of three ways:</p> <ul style="list-style-type: none"> • <u>idle timeout</u> (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>the first TCP-variant connection set up further includes, in addition to (a-1) sending the first TCP-variant packet that is also for use in determining the third duration of the third time period:</p>	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first TCP-variant connection (e.g., QUIC connection, etc.) set up further includes, in addition to (a-1) sending the first TCP-variant packet that is also for use in determining the third duration (e.g., duration associated with probe timeout, etc.) of the third time period (e.g., probe timeout, etc.):</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	(a-2) after sending the first TCP-variant packet and without any other communication between the first node and the	Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein, (a-2) after sending the first TCP-variant packet and without any other communication between the first node (e.g., one of a QUIC-compliant server or client, etc.) and the second node (e.g., the other one of the QUIC-compliant server or client, etc.) after (a-1) and before (a-2), receiving,

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	second node after (a-1) and before (a-2), receiving, at the first node from the second node, a second TCP-variant packet, and (a-3) after receiving the second TCP-variant packet and without any other communication between the first node and the second node after (a-2) and before (a-3), sending, from the first node to the second node, a third TCP-variant packet, and further wherein:	<p>at the first node from the second node, a second TCP-variant packet, and (a-3) after receiving the second TCP-variant packet and without any other communication between the first node and the second node after (a-2) and before (a-3), sending, from the first node to the second node, a third TCP-variant packet, and further wherein:</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <thead> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> </thead> <tbody> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </tbody> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																
	the first duration, the second duration, and the	Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	third duration are capable of being of different durations;	<p>duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), and the third duration (e.g., duration associated with probe timeout, etc.) are capable of being of different durations;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“<u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)”

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
	the second time period and the third time period are capable of being detected during the first TCP-variant connection set up and thereafter during the first TCP-variant connection; and	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) and the third time period (e.g., probe timeout, etc.) are capable of being detected during the first TCP-variant connection (e.g., QUIC connection, etc.) set up and thereafter during the first TCP-variant connection; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first time period and the second time period are started based on at least one of a received packet or a sent packet; and	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) and the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) are started based on at least one of a received packet or a sent packet; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, k\text{Granularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	during the at least portion of the first TCP-variant connection including the at least portion of the first TCP-variant connection set up, the detection of the first time period of the first	Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein, during the at least portion of the first TCP-variant connection (e.g., QUIC connection, etc.) including the at least portion of the first TCP-variant connection set up, the detection of the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), by being based on whether, during

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	<p>type of time period is based on the first duration, by being based on whether, during the first duration, no packet is received and processed to keep the first TCP-variant connection at least partially active; and</p>	<p>the first duration, no packet is received (e.g., connection remains idle, etc.) and processed to keep the first TCP-variant connection at least partially active; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: <u>original_connection_id</u>, <u>preferred_address</u>, <u>stateless_reset_token</u>, <u>ack_delay_exponent</u> and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
	based on the first duration being changed to the second duration, the detection of the second time period of the first type of time period is based on the second duration, by being based on whether, during the second duration, no packet is received and processed to keep the first TCP-variant connection at least partially active;	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein, based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) being changed to the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), the detection of the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the second duration, by being based on whether, during the second duration, no packet is received (e.g., connection remains idle, etc.) and processed to keep the first TCP-variant connection at least partially active;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	where only one of the first time period, the second time period, or the third time period is detected for the first TCP-variant connection.	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method where only one of the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.), the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.), or the third time period (e.g., probe timeout, etc.) is detected for the first TCP-variant connection (e.g., QUIC connection, etc.).</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
167	The method of claim 159, wherein: the first TCP-variant packet is not a synchronize (SYN) packet;	<p>Google infringes claim 159, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein: the first TCP-variant packet is not a synchronize (SYN) packet;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p><u>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p>

CLAIM CHARTS
BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: The following serves to distinguish the use of SYN handshake packets from QUIC negotiation packets where parameter values like idle_timeout may be exchanged.</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <p>1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y</p> <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. <u>The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN.</u> The three way handshake and the advantages of a clock-driven scheme are discussed in [3].”</p> <p>“Request for Comments” (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
	<p>the first TCP-variant connection is not a TCP extension connection that involves a TCP extension;</p>	<p>Google infringes claim 159, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first TCP-variant connection (e.g., QUIC connection, etc.) is not a TCP extension connection that involves a TCP extension;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since <u>QUIC is built on top of UDP</u>, it suffers from no such limitations.”</p> <p>https://www.chromium.org/quic</p>
	<p>the first type of time period includes an idle time period that does not include a user timeout period;</p>	<p>Google infringes claim 159, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first type of time period (e.g., idle-type timeout, etc.) includes an idle time period that does not include a user timeout period;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first time period is only capable of being detected during the first TCP-variant connection set up;	<p>Google infringes claim 159, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) is only capable of being detected during the first TCP-variant connection (e.g., QUIC connection, etc.) set up;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first duration is not negotiated between the first and second nodes for the first TCP-variant connection using (a-1), (a-2), nor (a-3), and the second duration and the third duration are negotiated between the first and second nodes for the first TCP-variant connection using at least	<p>Google infringes claim 159, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is not negotiated between the first and second nodes for the first TCP-variant connection (e.g., QUIC connection, etc.) using (a-1), (a-2), nor (a-3), and the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) and the third duration (e.g., duration associated with probe timeout, etc.) are negotiated between the first and second nodes for the first TCP-variant connection using at least one of (a-1), (a-2), or (a-3);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	one of (a-1), (a-2), or (a-3);	<p>Note: As set forth below, a <u>QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient</u> of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first duration is exclusively used for detecting the first type of time period until being changed to the second duration;	<p>Google infringes claim 159, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is exclusively used for detecting the first type of time period (e.g., idle-type timeout, etc.) until being changed to the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.);</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>the second duration is determined based on a first algorithm, and the third duration is determined based on a second algorithm that is different from the first algorithm;</p>	<p>Google infringes claim 159, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is determined based on a first algorithm (e.g., used to negotiate an idle timeout period between nodes, etc.), and the third duration (e.g., duration associated with probe timeout, etc.) is determined based on a second algorithm (e.g., used to determine the probe timeout, etc.) that is different from the first algorithm;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p><u>Note:</u> As set forth below, a <u>QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient</u> of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>$PTO = smoothed_rtt + \max(4 * rttvar, kGranularity) + \max_ack_delay$</p> <p>$kGranularity$, $smoothed_rtt$, $rttvar$, and \max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time ($smoothed_rtt$), the variance in the estimate ($4 * rttvar$), and \max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least $kGranularity$, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: Based upon information and belief, keep-alive signals are not required as a result of using idle timers and reducing the number of keep-alive signals to zero and, without a negotiated idle timeout, neither endpoint of a connection knows when the other side might timeout, so keep-alives (e.g., PINGs) have to be sent at unnecessarily shorter intervals to insure the connection to stay active. Further, with a negotiated idle time period, only one side sends one or more keep-alives, if a keep-alive is used.</p> <p>The QUIC IETF transport spec recognizes the need to avoid unnecessary keep-alives (e.g., PINGs). See section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2 and section 5.1 of the IETF HTTP/3 draft at https://tools.ietf.org/id/draft-ietf-quic-http-34.html#name-idle-connections.</p> <p>“19.2. PING Frame</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p>
	the third time period is detected based on first TCP-variant connection traffic-related criteria that only involves no acknowledgement packet being received in the first TCP-variant connection during the third duration and processed as an	<p>Google infringes claim 159, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the third time period (e.g., probe timeout, etc.) is detected based on first TCP-variant connection (e.g., QUIC connection, etc.) traffic-related criteria that only involves no acknowledgement packet being received in the first TCP-variant connection during the third duration (e.g., duration associated with probe timeout, etc.) and processed as an acknowledgement for a sent packet sent in the first TCP-variant connection;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	acknowledgement for a sent packet sent in the first TCP-variant connection;	<p>“18.1 Transport Parameter Definitions ...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>during the at least portion of the first TCP-variant connection including the at least portion of the first TCP-variant connection set up, the detection of the first time period of the first type of time period is based on the first duration, by being based on first TCP-variant connection traffic-related criteria that only involves whether any TCP-variant packet is received for the first TCP-variant connection set up during the first duration;</p>	<p>Google infringes claim 159, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein, during the at least portion of the first TCP-variant connection (e.g., QUIC connection, etc.) including the at least portion of the first TCP-variant connection set up, the detection of the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), by being based on first TCP-variant connection traffic-related criteria that only involves whether any TCP-variant packet is received for the first TCP-variant connection set up during the first duration;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	<p>based on the first duration being changed to the second duration, the detection of the second time period of the first type of time period is based on the second duration, by being based on first TCP-variant connection traffic-related criteria that only involves whether, during the second duration, no packet is received and processed to keep the first TCP-variant connection at least partially active, nor is expected to be received in the first TCP-variant connection; and</p>	<p>Google infringes claim 159, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein, based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) being changed to the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), the detection of the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the second duration, by being based on first TCP-variant connection (e.g., QUIC connection, etc.) traffic-related criteria that only involves whether, during the second duration, no packet is received (e.g., connection remains idle, etc.) and processed to keep the first TCP-variant connection at least partially active, nor is expected to be received in the first TCP-variant connection; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	an idle timeout attribute is utilized for the first type of time period, and a user timeout attribute, that is separate from the idle timeout attribute, is utilized for controlling communication of one or more acknowledgment packets.	<p>Google infringes claim 159, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein an idle timeout attribute is utilized for the first type of time period (e.g., idle-type timeout, etc.), and a user timeout attribute (e.g., acknowledgement in the form of PING frames, etc.), that is separate from the idle timeout attribute, is utilized for controlling communication of one or more acknowledgment packets.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“19.2. PING Frame</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer.</u> The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.” https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p>
168	The method of claim 167, wherein the third time period is detected independent of the detection of the first and second time periods.	<p>Google infringes claim 167, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the third time period (e.g., probe timeout, etc.) is detected independent of the detection of the first and second time periods.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions ...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate ($4 * \text{rttvar}$), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
169	The method of claim 167, wherein the second duration is the same as the first duration, despite being capable of being different from the first duration.	<p>Google infringes claim 167, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is the same as the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), despite being capable of being different from the first duration.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
170	The method of claim 78, wherein: the first TCP-variant packet is not a synchronize (SYN) packet;	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein: the first TCP-variant packet is not a synchronize (SYN) packet;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: The following serves to distinguish the use of SYN handshake packets from QUIC negotiation packets where parameter values like idle_timeout may be exchanged.</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“The synchronization requires each side to send its own initial sequence number and to receive a confirmation of it in acknowledgment from the other side. Each side must also receive the other side's initial sequence number and send a confirming acknowledgment.</p> <p>1) A --> B SYN my sequence number is X 2) A <-- B ACK your sequence number is X 3) A <-- B SYN my sequence number is Y 4) A --> B ACK your sequence number is Y</p> <p>Because steps 2 and 3 can be combined in a single message this is called the three way (or three message) handshake.</p> <p>A three way handshake is necessary because sequence numbers are not tied to a global clock in the network, and TCPs may have different mechanisms for picking the ISN's. <u>The receiver of the first SYN has no way of knowing whether the segment was an old delayed one or not, unless it remembers the last sequence number used on the connection (which is not always possible), and so it must ask the sender to verify this SYN.</u> The three way handshake and the advantages of a clock-driven scheme are discussed in [3].”</p> <p>“Request for Comments” (RFC) document RFC 793 (September 1981) https://tools.ietf.org/html/rfc793</p>
	<p>the first TCP-variant connection is not a TCP extension connection that involves a TCP extension;</p>	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first TCP-variant connection (e.g., QUIC connection, etc.) is not a TCP extension connection that involves a TCP extension;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: The following evidence is related to and/or describes the QUIC standard contained herein.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“On the surface, QUIC is very similar to TCP+TLS+HTTP/2 implemented on UDP. ...However, since <u>QUIC is built on top of UDP</u>, it suffers from no such limitations.”</p> <p>https://www.chromium.org/quic</p>
	<p>the first type of time period includes an idle time period that does not include a user timeout period;</p>	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first type of time period (e.g., idle-type timeout, etc.) includes an idle time period that does not include a user timeout period;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first time period is only capable of being detected during the first TCP-variant connection set up;	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) is only capable of being detected during the first TCP-variant connection (e.g., QUIC connection, etc.) set up;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first duration is not negotiated between the first and second nodes for the first TCP-variant connection using (a-1), (a-2), nor (a-3), and the second duration and the third duration are negotiated between the first and second nodes for the first TCP-variant connection using at least	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is not negotiated between the first and second nodes for the first TCP-variant connection (e.g., QUIC connection, etc.) using (a-1), (a-2), nor (a-3), and the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) and the third duration (e.g., duration associated with probe timeout, etc.) are negotiated between the first and second nodes for the first TCP-variant connection using at least one of (a-1), (a-2), or (a-3);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	one of (a-1), (a-2), or (a-3);	<p>Note: As set forth below, a <u>QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient</u> of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type <code>PROTOCOL_VIOLATION</code>.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the first duration is exclusively used for detecting the first type of time period until being changed to the second duration;	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is exclusively used for detecting the first type of time period (e.g., idle-type timeout, etc.) until being changed to the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.);</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	the second duration is determined based on a first algorithm, and the third duration is determined based on a second algorithm that is different from the first algorithm;	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is determined based on a first algorithm (e.g., used to negotiate an idle timeout period between nodes, etc.), and the third duration (e.g., duration associated with probe timeout, etc.) is determined based on a second algorithm (e.g., used to determine the probe timeout, etc.) that is different from the first algorithm;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p><u>Note:</u> As set forth below, a <u>QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient</u> of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>$PTO = smoothed_rtt + \max(4 * rttvar, kGranularity) + \max_ack_delay$</p> <p>$kGranularity$, $smoothed_rtt$, $rttvar$, and \max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time ($smoothed_rtt$), the variance in the estimate ($4 * rttvar$), and \max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least $kGranularity$, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: Based upon information and belief, keep-alive signals are not required as a result of using idle timers and reducing the number of keep-alive signals to zero and, without a negotiated idle timeout, neither endpoint of a connection knows when the other side might timeout, so keep-alives (e.g., PINGs) have to be sent at unnecessarily shorter intervals to insure the connection to stay active. Further, with a negotiated idle time period, only one side sends one or more keep-alives, if a keep-alive is used.</p> <p>The QUIC IETF transport spec recognizes the need to avoid unnecessary keep-alives (e.g., PINGs). See section 10.1.2 of the QUIC transport draft at https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34#section-10.1.2 and section 5.1 of the IETF HTTP/3 draft at https://tools.ietf.org/id/draft-ietf-quic-http-34.html#name-idle-connections.</p> <p>“19.2. PING Frame</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer. The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out. An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.”</p> <p>https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p>
	the third time period is detected based on first TCP-variant connection traffic-related criteria that only involves, during the third duration, no acknowledgement packet being received in the first TCP-variant connection and processed as an	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the third time period (e.g., probe timeout, etc.) is detected based on first TCP-variant connection (e.g., QUIC connection, etc.) traffic-related criteria that only involves, during the third duration (e.g., duration associated with probe timeout, etc.), no acknowledgement packet being received in the first TCP-variant connection and processed as an acknowledgement for a sent packet sent in the first TCP-variant connection;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	acknowledgement for a sent packet sent in the first TCP-variant connection;	<p>“18.1 Transport Parameter Definitions ...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2^{14} or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, k\text{Granularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>during the at least portion of the first TCP-variant connection including the at least portion of the first TCP-variant connection set up, the detection of the first time period of the first type of time period is based on the first duration, by being based on first TCP-variant connection traffic-related criteria that only involves whether any TCP-variant packet is received or sent for the first TCP-variant connection set up during the first duration;</p>	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein, during the at least portion of the first TCP-variant connection (e.g., QUIC connection, etc.) including the at least portion of the first TCP-variant connection set up, the detection of the first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.), by being based on first TCP-variant connection traffic-related criteria that only involves whether any TCP-variant packet is received or sent for the first TCP-variant connection set up during the first duration;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, a QUIC negotiation packet includes transport parameters that include an idle timeout parameter that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	<p>based on the first duration being changed to the second duration, the detection of the second time period of the first type of time period is based on the second duration, by being based on first TCP-variant connection traffic-related criteria that only involves whether, during the second duration, no packet is received and processed to keep the first TCP-variant connection at least partially active, sent, nor is expected to be received or sent in the first TCP-variant connection; and</p>	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein, based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) being changed to the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.), the detection of the second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.) is based on the second duration, by being based on first TCP-variant connection (e.g., QUIC connection, etc.) traffic-related criteria that only involves whether, during the second duration, no packet is received (e.g., connection remains idle, etc.) and processed to keep the first TCP-variant connection at least partially active, sent, nor is expected to be received or sent in the first TCP-variant connection; and</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters.</u> These declarations are made unilaterally by each endpoint. Endpoints</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.</p> <p>7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	an idle timeout attribute is utilized for the first type of time period, and a user timeout attribute, that is separate from the idle timeout attribute, is utilized for controlling communication of one or more acknowledgment packets.	<p>Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein an idle timeout attribute is utilized for the first type of time period (e.g., idle-type timeout, etc.), and a user timeout attribute (e.g., acknowledgement in the form of PING frames, etc.), that is separate from the idle timeout attribute, is utilized for controlling communication of one or more acknowledgment packets.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“19.2. PING Frame</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>Endpoints can use PING frames (type=0x01) to verify that their peers are still alive or to check reachability to the peer.</u> The PING frame contains no additional fields.</p> <p>The receiver of a PING frame simply needs to acknowledge the packet containing this frame.</p> <p><u>The PING frame can be used to keep a connection alive when an application or application protocol wishes to prevent the connection from timing out.</u> An application protocol SHOULD provide guidance about the conditions under which generating a PING is recommended. This guidance SHOULD indicate whether it is the client or the server that is expected to send the PING. Having both endpoints send PING frames without coordination can produce an excessive number of packets and poor performance.</p> <p>A connection will time out if no packets are sent or received for a period longer than the time negotiated using the max_idle_timeout transport parameter (see Section 10). However, state in middleboxes might time out earlier than that. Though REQ-5 in [RFC4787] recommends a 2 minute timeout interval, experience shows that sending packets every 15 to 30 seconds is necessary to prevent the majority of middleboxes from losing state for UDP flows.” https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27#section-19.2</p>
171	The method of claim 78, wherein the third time period is detected based on the third duration, by being detected when no acknowledgement packet is detected as being received and processed, during the third duration, by the first node in the first TCP-variant connection to	Google infringes claim 78, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the third time period (e.g., probe timeout, etc.) is detected based on the third duration (e.g., duration associated with probe timeout, etc.), by being detected when no acknowledgement packet is detected as being received and processed, during the third duration, by the first node (e.g., one of a QUIC-compliant server or client, etc.) in the first TCP-variant connection (e.g., QUIC connection, etc.) to acknowledge a sent packet of a plurality of sent packets sent by the first node, where the third duration starts for each of the plurality of sent packets when at least one of the plurality of sent packets is sent, and where a single timer and a single timeout variable are utilized for detecting the third time period in connection with the plurality of sent packets.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	<p>acknowledge a sent packet of a plurality of sent packets sent by the first node, where the third duration starts for each of the plurality of sent packets when at least one of the plurality of sent packets is sent, and where a single timer and a single timeout variable are utilized for detecting the third time period in connection with the plurality of sent packets.</p>	<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions ... • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> <p>PTO = smoothed_rtt + max(4*rttvar, kGranularity) + max_ack_delay</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time” <i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
172	The method of claim 171, wherein the third duration starts for each of the plurality of sent packets when the each one of the plurality of sent packets is sent.	<p>Google infringes claim 171, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the third duration (e.g., duration associated with probe timeout, etc.) starts for each of the plurality of sent packets when the each one of the plurality of sent packets is sent.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid”

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22
173	The method of claim 171, wherein the third duration starts for each of the plurality of sent packets when a first one of the plurality of sent packets is sent.	<p>Google infringes claim 171, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the third duration (e.g., duration associated with probe timeout, etc.) starts for each of the plurality of sent packets when a first one of the plurality of sent packets is sent.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quick-transport-22</p>
174	The method of claim 171, wherein the third duration starts for each of the plurality of sent packets when a last one of the plurality of sent packets is sent or received.	<p>Google infringes claim 171, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the third duration (e.g., duration associated with probe timeout, etc.) starts for each of the plurality of sent packets when a last one of the plurality of sent packets is sent or received.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt),</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>the variance in the estimate ($4 \cdot \text{rttvar}$), and <code>max_ack_delay</code>, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least <code>kGranularity</code>, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
175	<p>The method of claim 171, wherein the third duration starts for each of the plurality of sent packets when a first one of the plurality of sent packets is sent, where the first packet is a first packet to be received after a last one of the plurality of sent packets is sent.</p>	<p>Google infringes claim 171, through its products/services owns, controls, and/or uses at least one server computer and/or the Google Cloud Platform to perform a method wherein the third duration (e.g., duration associated with probe timeout, etc.) starts for each of the plurality of sent packets when a first one of the plurality of sent packets is sent, where the first packet is a first packet to be received after a last one of the plurality of sent packets is sent.</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“18.1 Transport Parameter Definitions ...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to 1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p> <p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

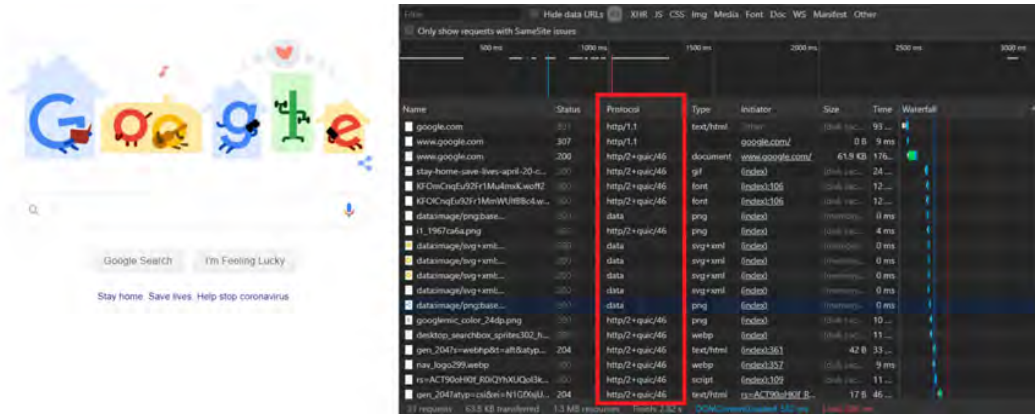
CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
176	An apparatus, comprising: a non-transitory memory means for storing instructions means; and one or more processors means in communication with the non-transitory memory means, wherein the one or more processors means execute the instructions means to:	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) comprising: a non-transitory memory means for storing instructions (e.g., server software, etc.) means; and one or more processors means in communication with the non-transitory memory means, wherein the one or more processors means execute the instructions means to:</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: Google admits that their “Google Cloud Platform” uses QUIC:</p> <p>"Google Cloud Platform Blog...</p> <p>we're happy to be the first major public cloud to offer QUIC support for our HTTPS load balancers."</p> <p>https://cloudplatform.googleblog.com/2018/06/Introducing-QUIC-support-for-HTTPS-load-balancing.html</p> <p>Note: The “Google Cloud Platform” that, as established above, uses QUIC, is also used for ALL of Google’s services:</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube”</p> <p>https://en.wikipedia.org/wiki/Google_Cloud_Platform</p> <p>Note: At least a portion of the citations herein are made to the QUIC standard found at: https://tools.ietf.org/html/draft-ietf-quic-transport-22, and/or https://tools.ietf.org/html/draft-ietf-quic-transport-27. Based upon information and belief, the pertinent portions of such version of the QUIC standard may also be found in one or more other versions. For completeness, all of the versions below (and future iterations thereof) are hereby incorporated by reference:</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		https://tools.ietf.org/html/draft-ietf-quic-transport-00 https://tools.ietf.org/html/draft-ietf-quic-transport-01 https://tools.ietf.org/html/draft-ietf-quic-transport-02 https://tools.ietf.org/html/draft-ietf-quic-transport-03 https://tools.ietf.org/html/draft-ietf-quic-transport-04 https://tools.ietf.org/html/draft-ietf-quic-transport-05 https://tools.ietf.org/html/draft-ietf-quic-transport-06 https://tools.ietf.org/html/draft-ietf-quic-transport-07 https://tools.ietf.org/html/draft-ietf-quic-transport-08 https://tools.ietf.org/html/draft-ietf-quic-transport-09 https://tools.ietf.org/html/draft-ietf-quic-transport-10 https://tools.ietf.org/html/draft-ietf-quic-transport-11 https://tools.ietf.org/html/draft-ietf-quic-transport-12 https://tools.ietf.org/html/draft-ietf-quic-transport-13 https://tools.ietf.org/html/draft-ietf-quic-transport-14 https://tools.ietf.org/html/draft-ietf-quic-transport-15 https://tools.ietf.org/html/draft-ietf-quic-transport-16 https://tools.ietf.org/html/draft-ietf-quic-transport-17 https://tools.ietf.org/html/draft-ietf-quic-transport-18 https://tools.ietf.org/html/draft-ietf-quic-transport-19 https://tools.ietf.org/html/draft-ietf-quic-transport-20 https://tools.ietf.org/html/draft-ietf-quic-transport-21 https://tools.ietf.org/html/draft-ietf-quic-transport-22 https://tools.ietf.org/html/draft-ietf-quic-transport-23 https://tools.ietf.org/html/draft-ietf-quic-transport-24 https://tools.ietf.org/html/draft-ietf-quic-transport-25 https://tools.ietf.org/html/draft-ietf-quic-transport-26 https://tools.ietf.org/html/draft-ietf-quic-transport-27

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>Note: Though these contentions may refer to a specific version of the QUIC standard, the presented evidence is not exclusive to this specific version and is compliant with and/or supported by current versions of the QUIC standard, including QUIC_VERSION_IETF_RFC_V1 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34), QUIC_VERSION_IETF_DRAFT_25 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-25), QUIC_VERSION_IETF_DRAFT_27 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-27), QUIC_VERSION_IETF_DRAFT_28 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-28), QUIC_VERSION_IETF_DRAFT_29 (https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-29), and/or QUIC_VERSION_50 which relies in part on IETF_QUIC_INVARIANTS_06 (https://tools.ietf.org/html/draft-ietf-quic-invariants-06) and/or IETF_QUIC_TRANSPORT_17 (https://tools.ietf.org/html/draft-ietf-quic-transport-17), each of which, upon information and belief, has been utilized by Google LLC through its accused instrumentalities.</p> <p>Note: Below is a web page of Google (https://www.google.com/).</p> 

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	identify, at a first node, first information on which at least a first duration for detecting a first type of time period is based;	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) configured to identify, at a first node (e.g., one of a QUIC-compliant server or client, etc.), first information (e.g., first value, etc.) on which at least a first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) for detecting a first type of time period (e.g., idle-type timeout, etc.) is based;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the idle_timeout field of the connection negotiation packet.</u></p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. Connection Termination</p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: <code>original_connection_id</code>, <code>preferred_address</code>, <code>stateless_reset_token</code>, <code>ack_delay_exponent</code> and <code>active_connection_id_limit</code>. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • <code>initial_max_data</code> • <code>initial_max_stream_data_bidi_local</code> • <code>initial_max_stream_data_bidi_remote</code> • <code>initial_max_stream_data_uni</code> • <code>initial_max_streams_bidi</code> • <code>initial_max_streams_uni</code> <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes <code>initial_max_data</code> and either <code>initial_max_streams_bidi</code> and</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“3.3. Session resumption versus Keep-alive</p> <p>Because QUIC is encapsulated in UDP, applications using QUIC must deal with short idle timeouts. Deployed stateful middleboxes will generally establish state for UDP flows on the first packet state, and keep state for much shorter idle periods than for TCP. According to a 2010 study ([Hatonen10]), UDP applications can assume that any NAT binding or other state entry will be expired after just thirty seconds of inactivity.</p> <p>A QUIC application has three strategies to deal with this issue:</p> <ul style="list-style-type: none"> • Ignore it, if the application-layer protocol consists only of interactions with no or very short idle periods. • Ensure there are no long idle periods.

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> <u>Resume the session after a long idle period, using 0-RTT resumption when appropriate.</u> <p>...</p> <p>Alternatively, the client (but not the server) can use session resumption instead of sending keepalive traffic. In this case, a client that wants to send data to a server over a connection idle longer than the server's idle timeout (available from the idle_timeout transport parameter) can simply reconnect. <u>When possible, this reconnection can use 0-RTT session resumption, reducing the latency involved with restarting the connection.</u> This of course only applies in cases in which 0-RTT data is safe, when the client is the restarting peer, and when the data to be sent is idempotent.”</p> <p>https://www.ietf.org/id/draft-ietf-quic-applicability-07.txt</p>
	allocate a first resource for a first non-Transmission Control Protocol (non-TCP) connection;	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) configured to allocate a first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) for a first non-Transmission Control Protocol (non-TCP) connection (e.g., QUIC connection, etc.);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“Google Cloud Platform (GCP), offered by Google (company), is a <u>suite of cloud computing services</u> that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube”</p> <p>https://en.wikipedia.org/wiki/Google_Cloud_Platform</p>
	generate a first non-TCP packet including a first parameter field identifying first metadata for use in determining a second duration for detecting the first type of time period;	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) configured to generate a first non-TCP packet (e.g., first QUIC negotiation packet, etc.) including a first parameter field (e.g., first idle_timeout parameter field, etc.) identifying first metadata (e.g., first idle timeout value, etc.) for use in determining a second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) for detecting the first type of time period (e.g., idle-type timeout, etc.);</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p>During connection establishment, <u>both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint.</u> Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18. Transport Parameter Encoding</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The format of the transport parameters is the TransportParameters struct from Figure 15. This is described using the presentation language from Section 3 of [TLS13].</p> <pre>enum { original_connection_id(0), idle_timeout(1), stateless_reset_token(2), max_packet_size(3), initial_max_data(4), initial_max_stream_data_bidi_local(5), initial_max_stream_data_bidi_remote(6), initial_max_stream_data_uni(7), initial_max_streams_bidi(8), initial_max_streams_uni(9), ack_delay_exponent(10), max_ack_delay(11), disable_migration(12), preferred_address(13), active_connection_id_limit(14), (65535) } TransportParameterId;</pre> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>Note: As set forth below, since the idle_timeout value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily modified based on the value received in the idle_timeout field of the connection negotiation packet.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	set up the first non-TCP connection, by sending, from the first node to a second node, the first non-TCP packet to provide the first metadata to the	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) configured to set up the first non-TCP connection (e.g., QUIC connection, etc.), by sending, from the first node (e.g., one of a QUIC-compliant server or client, etc.) to a second node (e.g., the other one of the QUIC-compliant server or client, etc.), the first non-TCP packet (e.g., first QUIC negotiation packet, etc.) to provide the first metadata (e.g., first idle timeout value, etc.) to the second node, for use by the second node in determining the</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	second node, for use by the second node in determining the second duration for detecting the first type of time period;	<p>second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) for detecting the first type of time period (e.g., idle-type timeout, etc.);</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, <u>a QUIC negotiation packet includes transport parameters that include an idle_timeout parameter</u> that is detected by a recipient of such packet.</p> <p>“7.3. Transport Parameters</p> <p><u>During connection establishment, both endpoints make authenticated declarations of their transport parameters. These declarations are made unilaterally by each endpoint. Endpoints are required to comply with the restrictions implied by these parameters; the description of each parameter includes rules for its handling.”</u></p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability																																																
		<table> <tr> <th>Value</th><th>Parameter Name</th><th>Specification</th></tr> <tr> <td>0x0000</td><td>original_connection_id</td><td>Section 18.1</td></tr> <tr> <td>0x0001</td><td>idle_timeout</td><td>Section 18.1</td></tr> <tr> <td>0x0002</td><td>stateless_reset_token</td><td>Section 18.1</td></tr> <tr> <td>0x0003</td><td>max_packet_size</td><td>Section 18.1</td></tr> <tr> <td>0x0004</td><td>initial_max_data</td><td>Section 18.1</td></tr> <tr> <td>0x0005</td><td>initial_max_stream_data_bidi_local</td><td>Section 18.1</td></tr> <tr> <td>0x0006</td><td>initial_max_stream_data_bidi_remote</td><td>Section 18.1</td></tr> <tr> <td>0x0007</td><td>initial_max_stream_data_uni</td><td>Section 18.1</td></tr> <tr> <td>0x0008</td><td>initial_max_streams_bidi</td><td>Section 18.1</td></tr> <tr> <td>0x0009</td><td>initial_max_streams_uni</td><td>Section 18.1</td></tr> <tr> <td>0x000a</td><td>ack_delay_exponent</td><td>Section 18.1</td></tr> <tr> <td>0x000b</td><td>max_ack_delay</td><td>Section 18.1</td></tr> <tr> <td>0x000c</td><td>disable_migration</td><td>Section 18.1</td></tr> <tr> <td>0x000d</td><td>preferred_address</td><td>Section 18.1</td></tr> <tr> <td>0x000e</td><td>active_connection_id_limit</td><td>Section 18.1</td></tr> </table> <p style="text-align: center;">Table 6: Initial QUIC Transport Parameters Entries</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>	Value	Parameter Name	Specification	0x0000	original_connection_id	Section 18.1	0x0001	idle_timeout	Section 18.1	0x0002	stateless_reset_token	Section 18.1	0x0003	max_packet_size	Section 18.1	0x0004	initial_max_data	Section 18.1	0x0005	initial_max_stream_data_bidi_local	Section 18.1	0x0006	initial_max_stream_data_bidi_remote	Section 18.1	0x0007	initial_max_stream_data_uni	Section 18.1	0x0008	initial_max_streams_bidi	Section 18.1	0x0009	initial_max_streams_uni	Section 18.1	0x000a	ack_delay_exponent	Section 18.1	0x000b	max_ack_delay	Section 18.1	0x000c	disable_migration	Section 18.1	0x000d	preferred_address	Section 18.1	0x000e	active_connection_id_limit	Section 18.1
Value	Parameter Name	Specification																																																
0x0000	original_connection_id	Section 18.1																																																
0x0001	idle_timeout	Section 18.1																																																
0x0002	stateless_reset_token	Section 18.1																																																
0x0003	max_packet_size	Section 18.1																																																
0x0004	initial_max_data	Section 18.1																																																
0x0005	initial_max_stream_data_bidi_local	Section 18.1																																																
0x0006	initial_max_stream_data_bidi_remote	Section 18.1																																																
0x0007	initial_max_stream_data_uni	Section 18.1																																																
0x0008	initial_max_streams_bidi	Section 18.1																																																
0x0009	initial_max_streams_uni	Section 18.1																																																
0x000a	ack_delay_exponent	Section 18.1																																																
0x000b	max_ack_delay	Section 18.1																																																
0x000c	disable_migration	Section 18.1																																																
0x000d	preferred_address	Section 18.1																																																
0x000e	active_connection_id_limit	Section 18.1																																																

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“QUIC provides reliable, ordered delivery of the cryptographic handshake data. QUIC packet protection is used to encrypt as much of the handshake protocol as possible. The cryptographic handshake MUST provide the following properties:</p> <ul style="list-style-type: none"> • authenticated key exchange, where <ul style="list-style-type: none"> ○ a server is always authenticated, ○ a client is optionally authenticated, ○ every connection produces distinct and unrelated keys, ○ keying material is usable for packet protection for both 0-RTT and 1-RTT packets, and ○ 1-RTT keys have forward secrecy • <u>authenticated values for the transport parameters of the peer</u> (see Section 7.3) • authenticated negotiation of an application protocol (TLS uses ALPN [RFC7301] for this purpose)” <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	<p>in response to detecting, based on the first duration and by the first node during at least a portion of the first non-TCP connection including at least a portion of the first non-TCP connection set up, a first time period of the first type of time period, at least partially close the first non-TCP</p>	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) configured to, in response to detecting, based on the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) and by the first node (e.g., one of a QUIC-compliant server or client, etc.) during at least a portion of the first non-TCP connection (e.g., QUIC connection, etc.) including at least a portion of the first non-TCP connection set up, a first time period (e.g., idle timeout in connection with 0-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.), at least partially close (e.g., terminate, etc.) the first non-TCP connection, by releasing, by the first node, the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) allocated for the first non-TCP connection;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	connection, by releasing, by the first node, the first resource allocated for the first non-TCP connection;	<p>Note: As set forth below, since the <u>idle_timeout</u> value sets the duration of idleness, after which the connection is shutdown, a timeout attribute of the connection is necessarily detected based on the value received in the <u>idle_timeout</u> field of the connection negotiation packet.</p> <p>“idle_timeout (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC <u>connection can be terminated</u> in one of three ways:</p> <ul style="list-style-type: none"> • <u>idle timeout</u> (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout</u> (see Section 18.1) and three times the current Probe Timeout (PTO).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p> <p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<ul style="list-style-type: none"> • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	in response to detecting, based on the second duration and by the first node after the first duration is changed to the	Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) configured to, in response to detecting, based on the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) and by the first node (e.g., one of a QUIC-compliant server or client, etc.) after the first duration (e.g., duration associated with idle timeout in connection with 0-RTT packets, etc.) is changed to

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	second duration, a second time period of the first type of time period, at least partially close the first non-TCP connection, by releasing, by the first node, the first resource allocated for the first non-TCP connection;	<p>the second duration, a second time period (e.g., idle timeout in connection with 1-RTT packets, etc.) of the first type of time period (e.g., idle-type timeout, etc.), at least partially close (e.g., terminate, etc.) the first non-TCP connection (e.g., QUIC connection, etc.), by releasing, by the first node, the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) allocated for the first non-TCP connection;</p> <p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>Note: As set forth below, since the <code>idle_timeout</code> value sets the duration of idleness, after which the connection is shutdown, <u>a timeout attribute of the connection is necessarily detected based on the value received in the <code>idle_timeout</code> field of the connection negotiation packet.</u></p> <p>“<code>idle_timeout</code> (0x0001): The idle timeout is a value in milliseconds that is encoded as an integer; see (Section 10.2). If this parameter is absent or zero then the idle timeout is disabled.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“10. <u>Connection Termination</u></p> <p>An established QUIC connection can be terminated in one of three ways:</p> <ul style="list-style-type: none"> • idle timeout (Section 10.2) • immediate close (Section 10.3) • stateless reset (Section 10.4) <p>An endpoint MAY discard connection state if it does not have a validated path on which it can send packets (see Section 8.2).”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>“10.2. Idle Timeout</p> <p>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p><u>The value for an idle timeout can be asymmetric.</u> The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“7.3.1. Values of Transport Parameters for 0-RTT</p> <p><u>Both endpoints store the value of the server transport parameters from a connection and apply them to any 0-RTT packets that are sent in subsequent connections to that peer, except for transport parameters that are explicitly excluded. Remembered transport parameters apply to the new connection until the handshake completes and the client starts sending 1-RTT packets.</u> Once the handshake completes, the client uses the transport parameters established in the handshake.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The definition of new transport parameters (Section 7.3.2) MUST specify whether they MUST, MAY, or MUST NOT be stored for 0-RTT. A client need not store a transport parameter it cannot process.</p> <p>A client MUST NOT use remembered values for the following parameters: original_connection_id, preferred_address, stateless_reset_token, ack_delay_exponent and active_connection_id_limit. The client MUST use the server's new values in the handshake instead, and absent new values from the server, the default value.</p> <p>A client that attempts to send 0-RTT data MUST remember all other transport parameters used by the server. The server can remember these transport parameters, or store an integrity-protected copy of the values in the ticket and recover the information when accepting 0-RTT data. A server uses the transport parameters in determining whether to accept 0-RTT data.</p> <p>If 0-RTT data is accepted by the server, the server MUST NOT reduce any limits or alter any values that might be violated by the client with its 0-RTT data. In particular, a server that accepts 0-RTT data MUST NOT set values for the following parameters (Section 18.1) that are smaller than the remembered value of the parameters.</p> <ul style="list-style-type: none"> • initial_max_data • initial_max_stream_data_bidi_local • initial_max_stream_data_bidi_remote • initial_max_stream_data_uni • initial_max_streams_bidi • initial_max_streams_uni <p>Omitting or setting a zero value for certain transport parameters can result in 0-RTT data being enabled, but not usable. The applicable subset of transport parameters that permit sending of application data SHOULD be set to non-zero values for 0-RTT. This includes</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>initial_max_data and either initial_max_streams_bidi and initial_max_stream_data_bidi_remote, or initial_max_streams_uni and initial_max_stream_data_uni.</p> <p>A server MUST either reject 0-RTT data or abort a handshake if the implied values for transport parameters cannot be supported.</p> <p>When sending frames in 0-RTT packets, a client MUST only use remembered transport parameters; importantly, it MUST NOT use updated values that it learns from the server's updated transport parameters or from frames received in 1-RTT packets. Updated values of transport parameters from the handshake apply only to 1-RTT packets. For instance, flow control limits from remembered transport parameters apply to all 0-RTT packets even if those values are increased by the handshake or by frames sent in 1-RTT packets. A server MAY treat use of updated transport parameters in 0-RTT as a connection error of type PROTOCOL_VIOLATION.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p>
	in response to detecting, based on a third duration and by the first node, a third time period, at least partially close the first non-TCP connection, by releasing, by the first node, the first resource allocated for the first non-TCP connection, where the third duration is determined based on a first algorithm that is different	<p>Google uses an apparatus (e.g., a platform or device, including a QUIC-compliant server or client, etc.) configured to, in response to detecting, based on a third duration (e.g., duration associated with probe timeout, etc.) and by the first node (e.g., one of a QUIC-compliant server or client, etc.), a third time period (e.g., probe timeout, etc.), at least partially close (e.g., terminate, etc.) the first non-TCP connection (e.g., QUIC connection, etc.), by releasing, by the first node, the first resource (e.g. a storage and/or data structure for storing connection IDs, a connection state, send/recv buffers, etc.) allocated for the first non-TCP connection, where the third duration is determined based on a first algorithm (e.g., used to determine the probe timeout, etc.) that is different from a second algorithm (e.g., used to negotiate an idle timeout period between nodes, etc.) on which a determination of the second duration (e.g., duration associated with idle timeout in connection with 1-RTT packets, etc.) is based.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
	from a second algorithm on which a determination of the second duration is based.	<p>See excerpt(s) above and below, for example (emphasis added, if any):</p> <p>“10.2. Idle Timeout</p> <p><u>If the idle timeout is enabled, a connection is silently closed and the state is discarded when it remains idle for longer than both the advertised idle timeout (see Section 18.1) and three times the current Probe Timeout (PTO).</u></p> <p>Each endpoint advertises its own idle timeout to its peer. An endpoint restarts any timer it maintains when a packet from its peer is received and processed successfully. The timer is also restarted when sending a packet containing frames other than ACK or PADDING (an ACK-eliciting packet; see [QUIC-RECOVERY]), but only if no other ACK-eliciting packets have been sent since last receiving a packet. Restarting when sending packets ensures that connections do not prematurely time out when initiating new activity.</p> <p>The value for an idle timeout can be asymmetric. The value advertised by an endpoint is only used to determine whether the connection is live at that endpoint. An endpoint that sends packets near the end of the idle timeout period of a peer risks having those packets discarded if its peer enters the draining state before the packets arrive. If a peer could timeout within a Probe Timeout (PTO; see Section 6.3 of [QUIC-RECOVERY]), it is advisable to test for liveness before sending any data that cannot be retried safely. Note that it is likely that only applications or application protocols will know what information can be retried.”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“18.1 Transport Parameter Definitions</p> <p>...</p> <ul style="list-style-type: none"> • max_ack_delay (0x000b): The maximum ACK delay is an integer value indicating the maximum amount of time in milliseconds by which the endpoint will delay sending acknowledgments. This value SHOULD include the receiver's expected delays in alarms firing. For example, if a receiver sets a timer for 5ms and alarms commonly fire up to

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>1ms late, then it should send a max_ack_delay of 6ms. If this value is absent, a default of 25 milliseconds is assumed. Values of 2¹⁴ or greater are invalid”</p> <p><i>QUIC: A UDP-Based Multiplexed and Secure Transport</i> https://tools.ietf.org/html/draft-ietf-quic-transport-22</p> <p>“6.3. Probe Timeout</p> <p>A Probe Timeout (PTO) triggers a probe packet when ack-eliciting data is in flight but an acknowledgement is not received within the expected period of time. A PTO enables a connection to recover from loss of tail packets or acks. The PTO algorithm used in QUIC implements the reliability functions of Tail Loss Probe [TLP] [RACK], RTO [RFC5681] and F-RTO algorithms for TCP [RFC5682], and the timeout computation is based on TCP's retransmission timeout period [RFC6298].</p> <p>6.3.1. Computing PTO</p> <p>When an ack-eliciting packet is transmitted, the sender schedules a timer for the PTO period as follows:</p> $\text{PTO} = \text{smoothed_rtt} + \max(4 * \text{rttvar}, \text{kGranularity}) + \text{max_ack_delay}$ <p>kGranularity, smoothed_rtt, rttvar, and max_ack_delay are defined in Appendix A.2 and Appendix A.3.</p> <p>The PTO period is the amount of time that a sender ought to wait for an acknowledgement of a sent packet. This time period includes the estimated network roundtrip-time (smoothed_rtt), the variance in the estimate (4*rttvar), and max_ack_delay, to account for the maximum time by which a receiver might delay sending an acknowledgement.</p>

CLAIM CHARTS
 BASED ON INFRINGEMENT ANALYSIS OF GOOGLE
 U.S. Patent No. 10,951,742

Claim	Claim Elements	Applicability
		<p>The PTO value MUST be set to at least kGranularity, to avoid the timer expiring immediately.</p> <p>When a PTO timer expires, the sender probes the network as described in the next section. The PTO period MUST be set to twice its current value. This exponential reduction in the sender's rate is important because the PTOs might be caused by loss of packets or acknowledgements due to severe congestion.</p> <p>A sender computes its PTO timer every time an ack-eliciting packet is sent. A sender might choose to optimize this by setting the timer fewer times if it knows that more ack-eliciting packets will be sent within a short period of time”</p> <p><i>QUIC Loss Detection and Congestion Control</i> https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-22#section-6.3</p>

Caveat: The notes and/or cited excerpts utilized herein are set forth for illustrative purposes only and are not meant to be limiting in any manner. For example, the notes and/or cited excerpts, may or may not be supplemented or substituted with different excerpt(s) of the relevant reference(s), as appropriate. Further, to the extent any error(s) and/or omission(s) exist herein, all rights are reserved to correct the same.

EXHIBIT 19

**UNITED STATES
PATENT AND TRADEMARK OFFICE**



PTAB Trial Statistics FY21 Q3 Outcome Roundup IPR, PGR, CBM

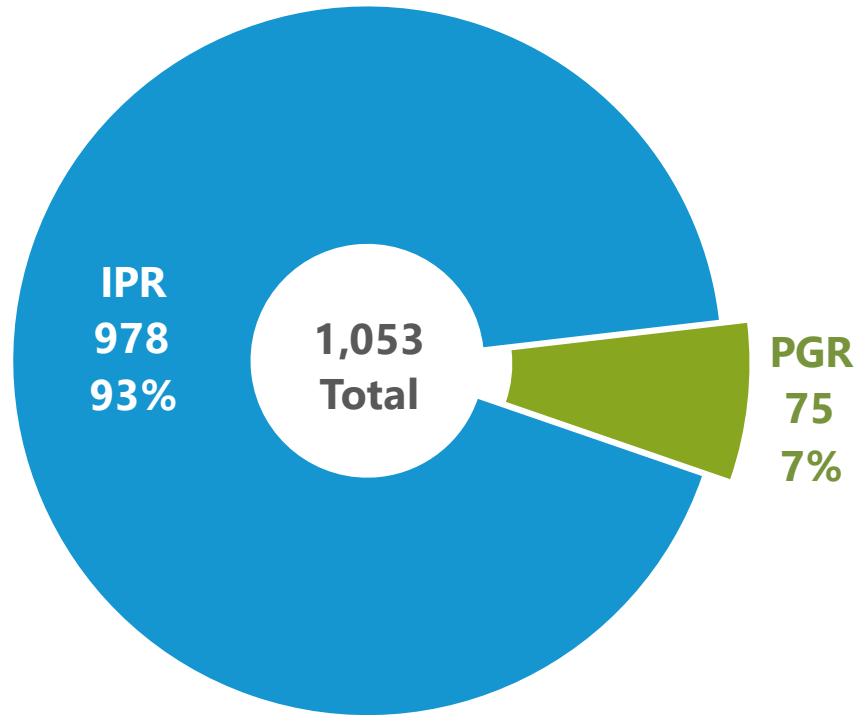
Patent Trial and Appeal Board
Fiscal Year 2021 3rd Quarter

UNITED STATES
PATENT AND TRADEMARK OFFICE



Petitions filed by trial type

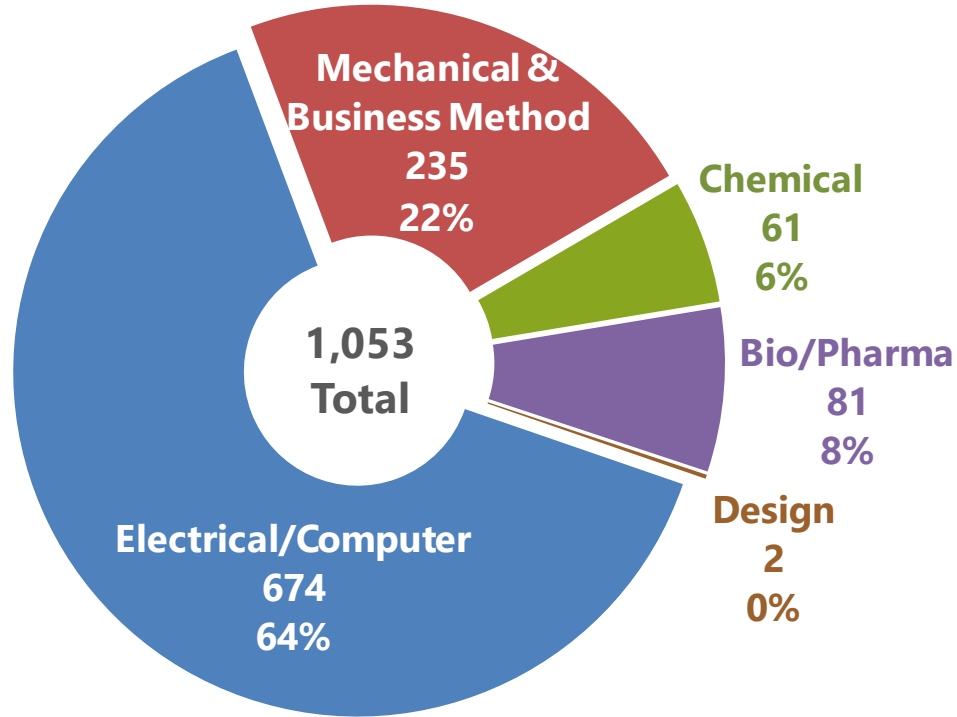
(FY21 through Q3: Oct. 1, 2020 to Jun. 30, 2021)



Trial types include Inter Partes Review (IPR), Post Grant Review (PGR), and Covered Business Method (CBM). The Office will not consider a CBM petition filed on or after September 16, 2020.

Petitions filed by technology

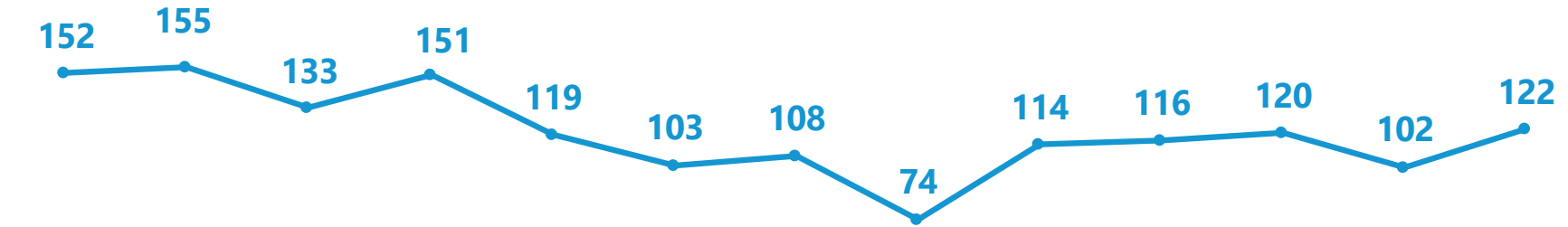
(FY21 through Q3: Oct. 1, 2020 to Jun. 30, 2021)



Petitions filed by month

(Jun. 2021 and Previous 12 Months: Jun. 1, 2020 to Jun. 30, 2021)

(978 IPRs in FY21)



Jun-20

IPR

Jun-21

(75 PGRs in FY21)

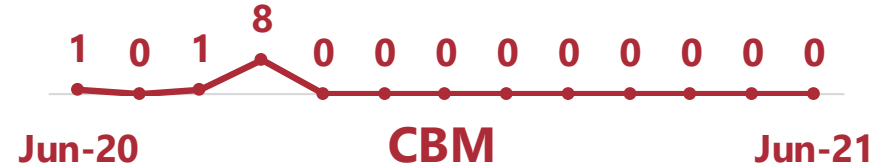


Jun-20

PGR

Jun-21

(0 CBMs in FY21)



Jun-20

CBM

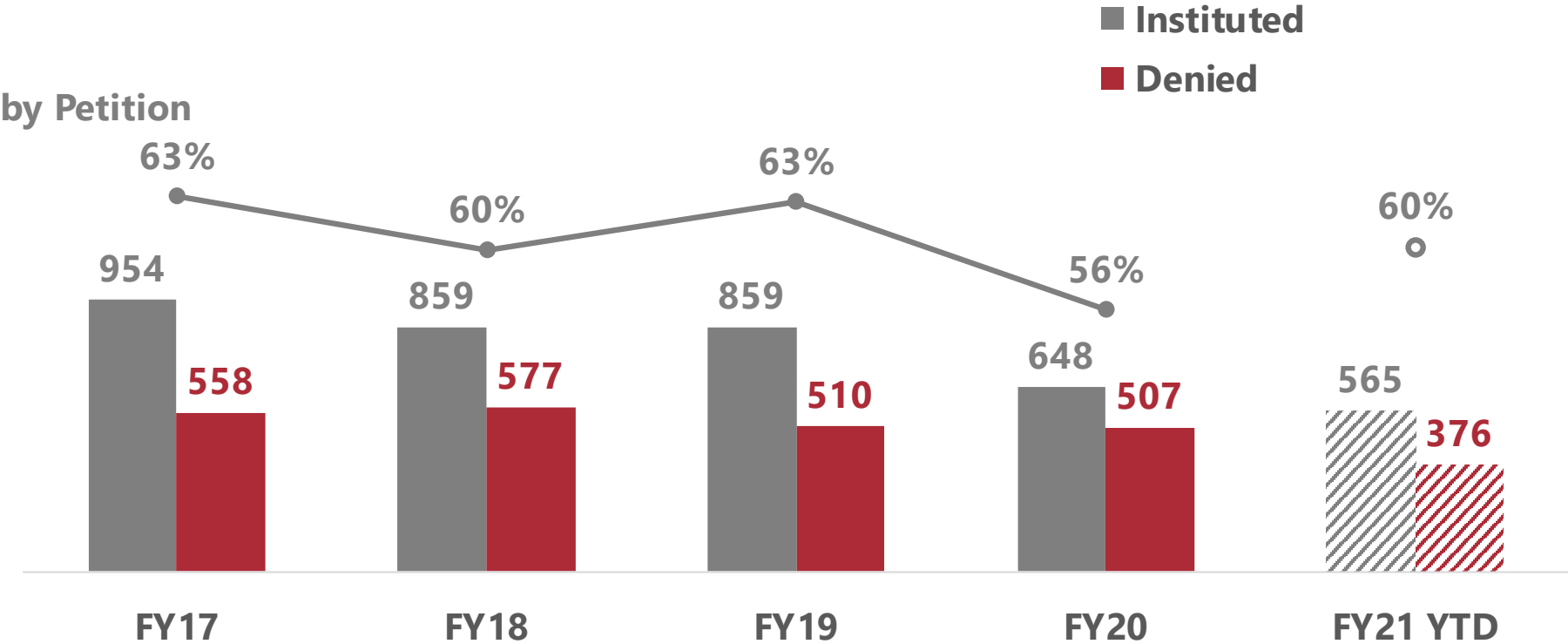
Jun-21



The Office will not consider a CBM petition filed on or after September 16, 2020.

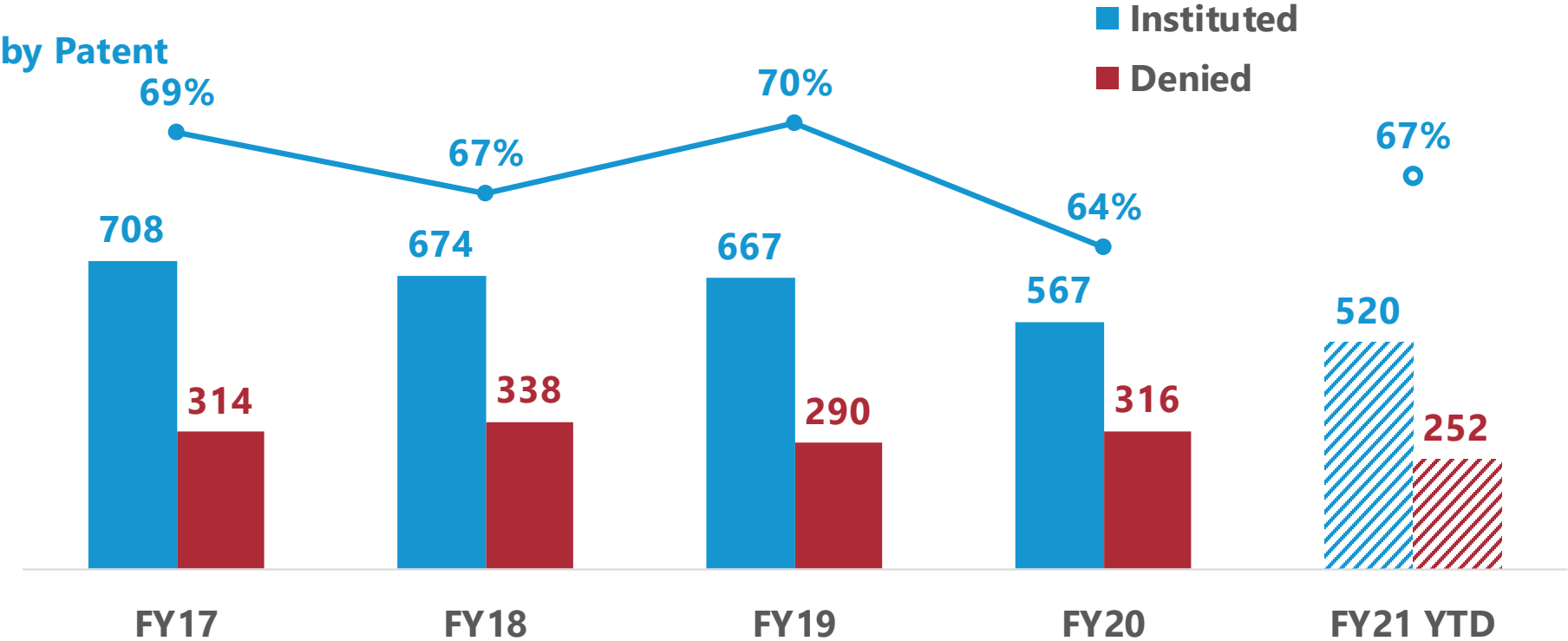
Institution rates by petition

(FY17 to FY21 through Q3: Oct. 1, 2016 to Jun. 30, 2021)



Institution rates by patent

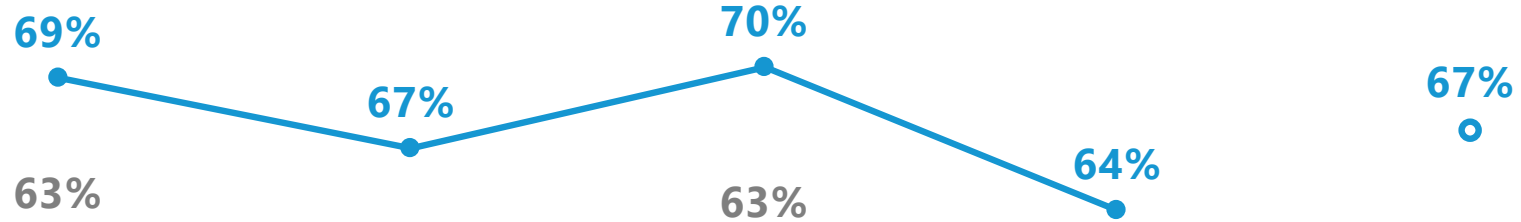
(FY17 to FY21 through Q3: Oct. 1, 2016 to Jun. 30, 2021)



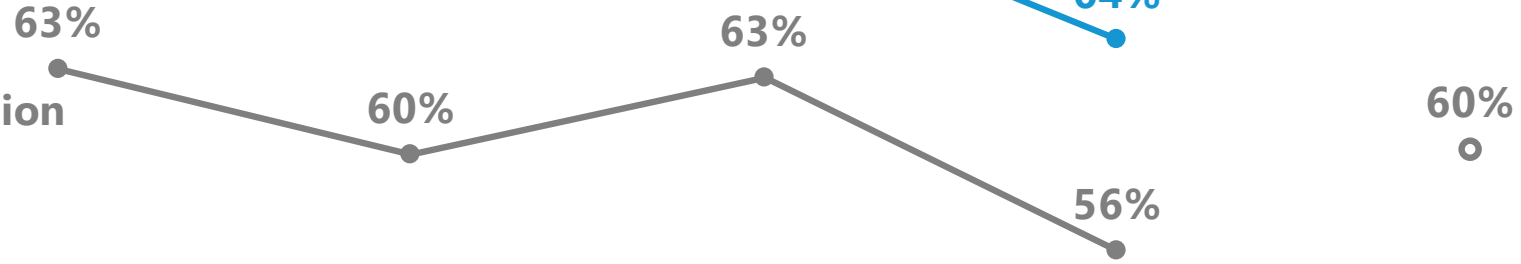
Institution rates by patent and by petition

(FY17 to FY21 through Q3: Oct. 1, 2016 to Jun. 30, 2021)

by Patent



by Petition



FY17

FY18

FY19

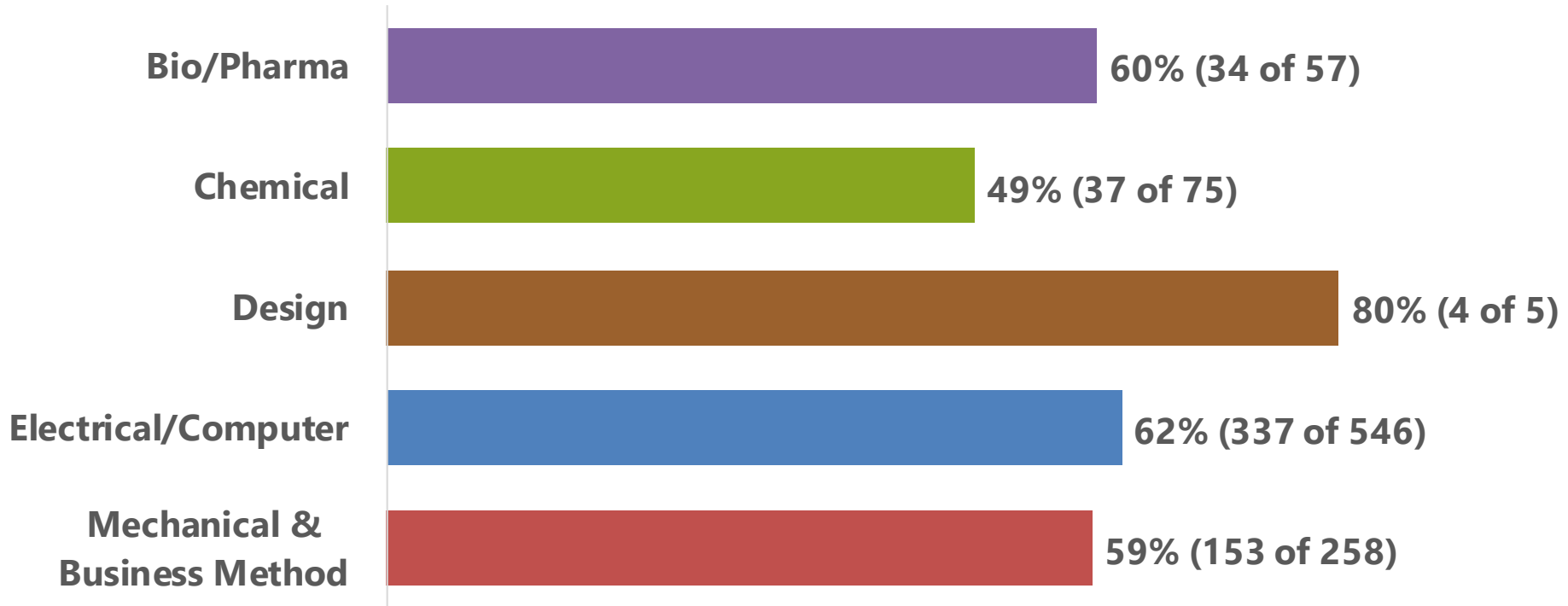
FY20

FY21 YTD

uspto

Institution rates by technology

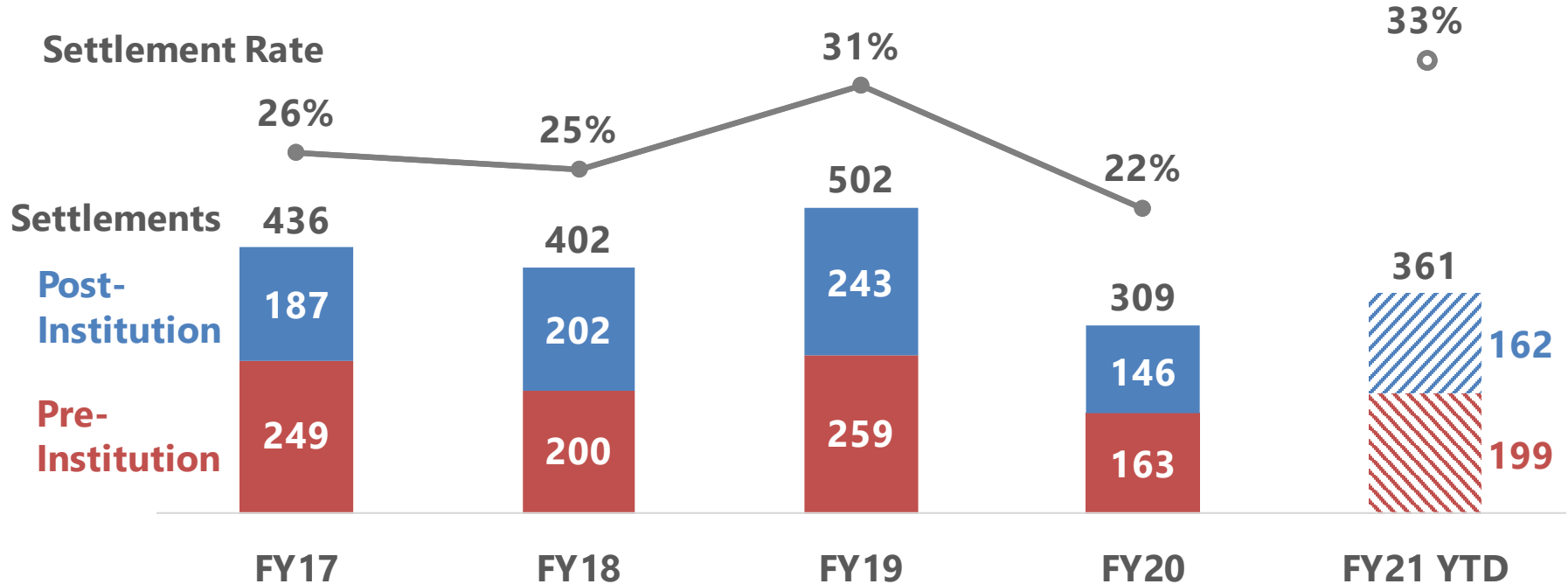
(FY21 through Q3: Oct. 1, 2020 to Jun. 30, 2021)



Institution rate for each technology is calculated by dividing petitions instituted by decisions on institution (i.e., petitions instituted plus petitions denied). The outcomes of decisions on institution responsive to requests for rehearing are excluded.

Settlements

(FY17 to FY21 through Q3: Oct. 1, 2016 to Jun. 30, 2021)

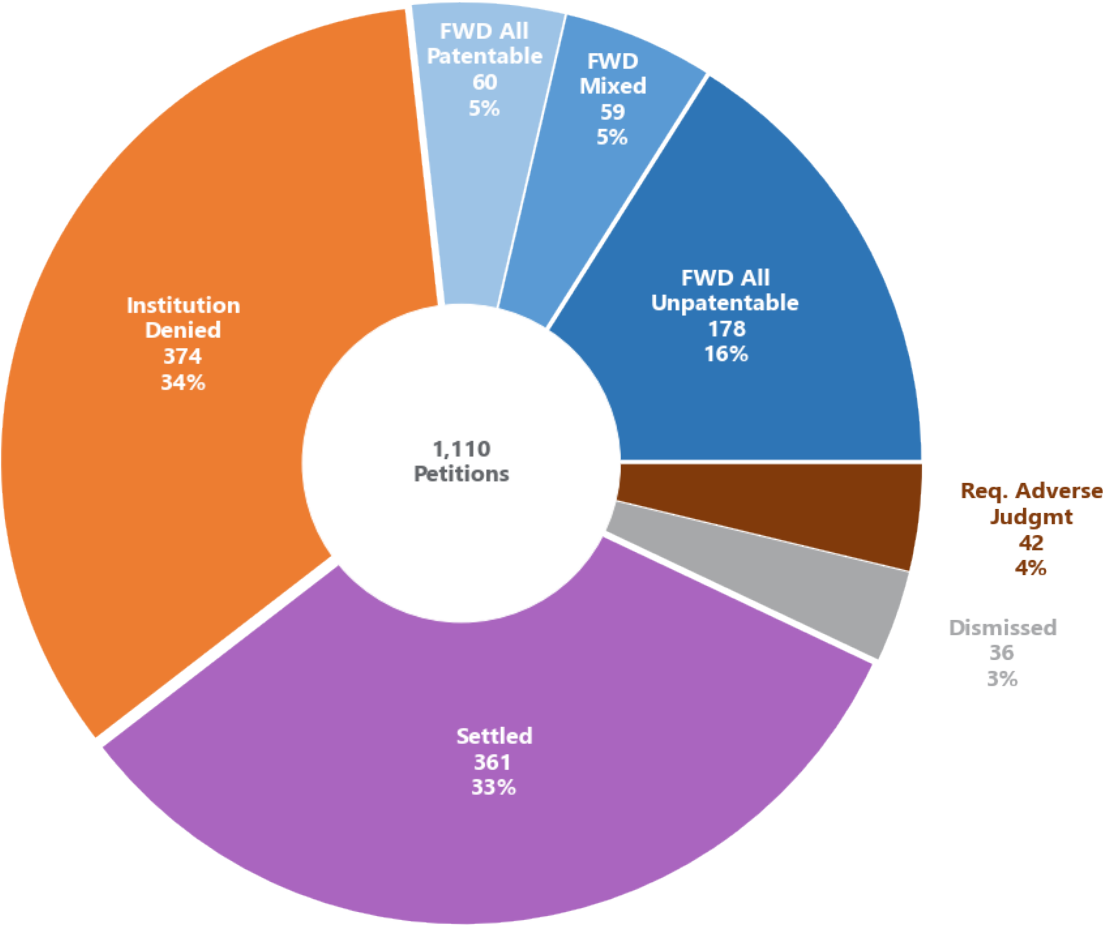


Settlement rate is calculated by dividing total settlements by concluded proceedings in each fiscal year (i.e., denied institution, settled, dismissed, requested adverse judgment, and final written decision), excluding joined cases.



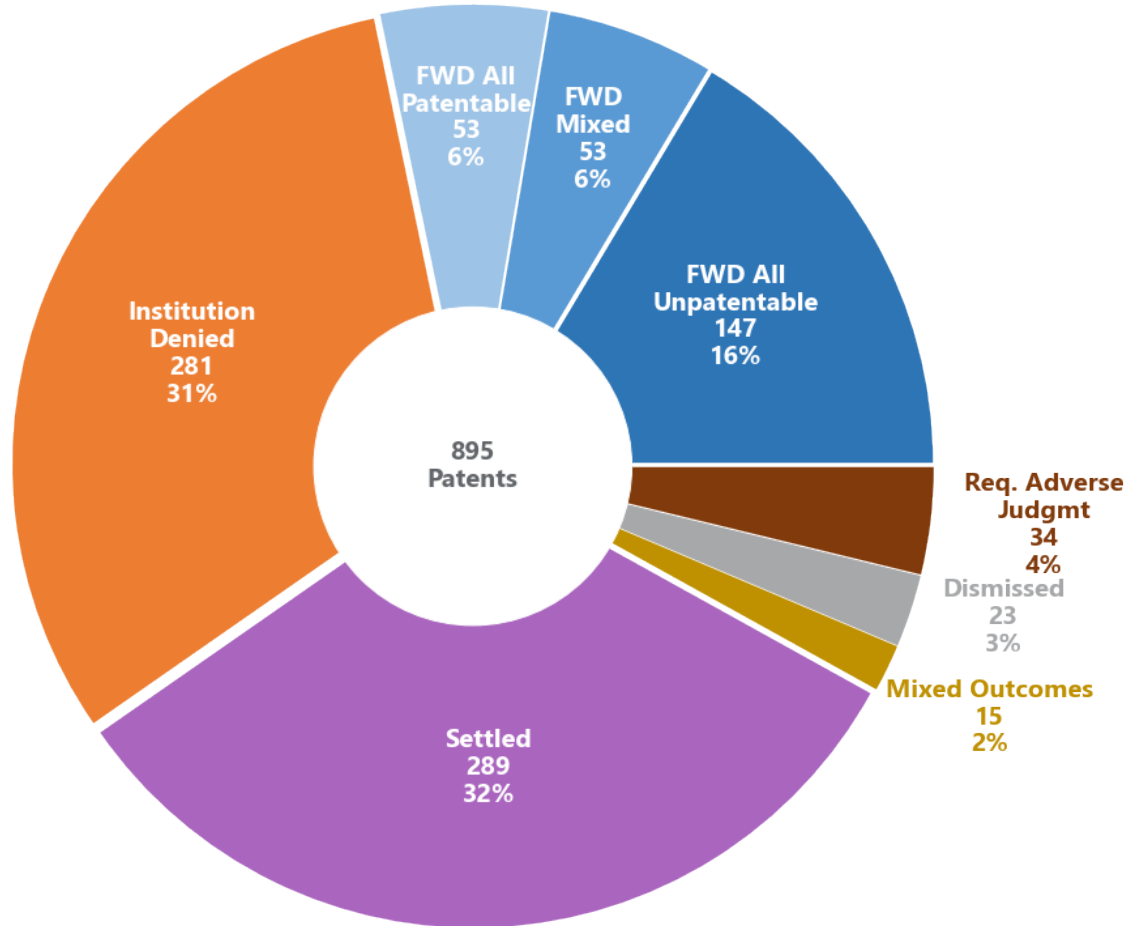
Outcomes by petition

(FY21 through Q3: Oct. 1, 2020 to Jun. 30, 2021)



Outcomes by patent

(FY21 through Q3: Oct. 1, 2020 to Jun. 30, 2021)



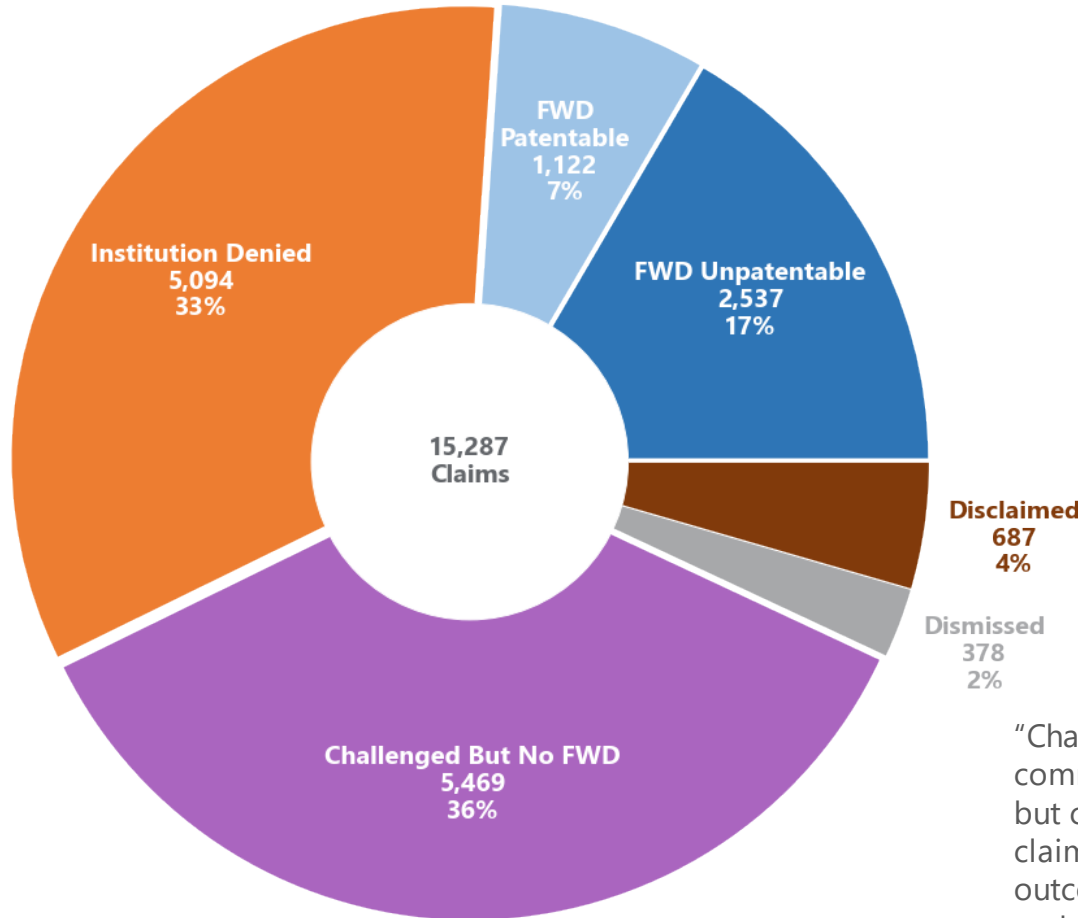
FWD patentability or unpatentability reported with respect to the claims at issue in the FWD.

"Mixed Outcome" is shown for patents receiving more than one type of outcome from the list of: denied, settled, dismissed, and/or req. adverse judgement only.

A patent is listed in a FWD category if it ever received a FWD, regardless of other outcomes.

Outcomes by claim challenged

(FY21 through Q3: Oct. 1, 2020 to Jun. 30, 2021)

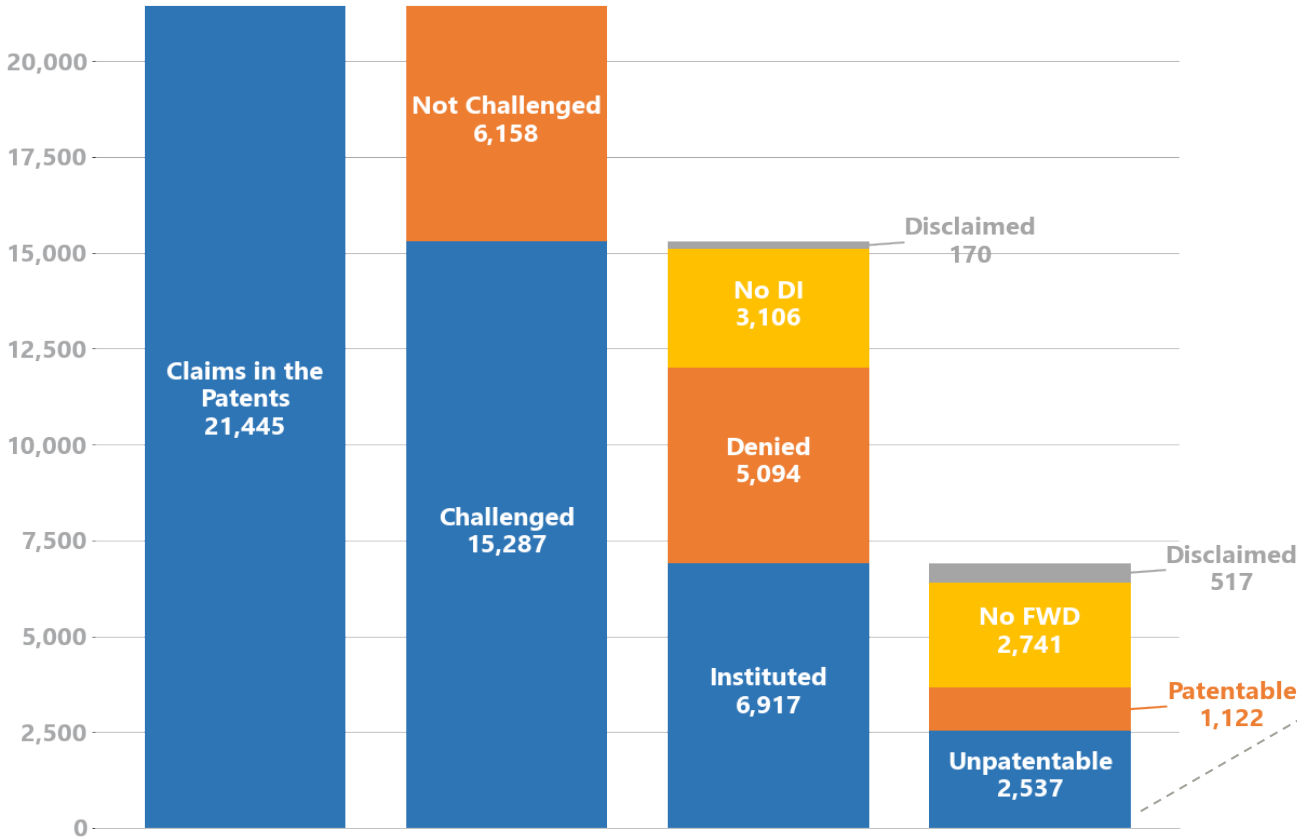


"Challenged But No FWD" most commonly refers to settlement, but can also refer to challenged claims subject to mixed non-FWD outcomes, such as both a denial and a dismissal.



Claim outcomes

(FY21 through Q3: Oct. 1, 2020 to Jun. 30, 2021)



*"No DI" and "No FWD" means the claim was challenged but not addressed in a DI/FWD, e.g., due to settlement.

17% of challenged claims and 37% of instituted claims were found unpatentable by a preponderance of the evidence.





EXHIBIT 20



United States Patent and Trademark Office

Commissioner for Patents

United States Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2020

1. Total requests filed since start of <i>Ex Parte</i> reexam on 07/01/81 ¹	14185
a. By patent owner	3954 27.9%
b. by other member of the public	10056 70.9%
c. By order of Commissioner	175 1.2%
2. Number of Filings by discipline	
a. Chemical Operation	3843 27.1%
b. Electrical Operation	5279 37.2%
c. Mechanical Operation	4790 33.8%
d. Design Patents	273 1.9%
3. Annual <i>Ex Parte</i> Reexam Filings	
Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No	
1981 78 (3 mos.) 1989 242 1997 380 2005 520 2013 305	
1982 187 1990 297 1998 348 2006 473 2014 355	
1983 186 1991 309 1999 391 2007 600 2015 244	
1984 189 1992 390 2000 312 2008 690 2016 222	
1985 230 1993 359 2001 295 2009 603 2017 191	
1986 232 1994 379 2002 272 2010 697 2018 187	
1987 240 1995 395 2003 394 2011 734 2019 163	
1988 269 1996 413 2004 436 2012 780 2020 198	
4. Number known to be in litigation.....	5428 38.3%
5. Decisions on requests ²	14185
a. No. granted	13082 92.2%
(1) By examiner	12888
(2) By Director (on petition)	194
b. No. denied	1183 8.3%
(1) By examiner	1103
(2) Reexam vacated	80
6. Total examiner denials (includes denials reversed by Director).....	1103
a. Patent owner requester	525 47.6%
b. Third party requester	578 52.4%

¹ Total decisions on requests does not include requests that have been vacated or are pending.² The Director granted requests are only granted petitions on denial under 37 CFR 1.515.³ Historic certificates were reviewed and the data in items 8-10 reflects the outcome of the review to the extent the data is available as of Oct 1, 2020

7. Overall reexamination pendency (Filing date to certificate issue date).....	
a. Average pendency	25.7 (mos.)
b. Median pendency	19.3 (mos.)

8. Reexam certificate claim analysis: ³	Owner Requester	3rd Party Requester	Comm'r Initiate	Overall	Numbers
a. Certificates with all claims confirmed	6%	14%	0%	20.9%	2672
b. Certificates with all claims canceled	3%	10%	0%	13.1%	1668
c. Certificates with claims changes	22%	44%	1%	66.0%	8424

9. Total ex parte reexamination certificates issued (1981 – present) 12764

10. Reexam claim analysis – requester is patent owner or 3rd party or Commissioner initiated.

a. Certificates – PATENT OWNER REQUESTER		3960
a. All claims confirmed	809	20.4%
b. All claims canceled	396	10.0%
c. Claims changed	2755	69.6%
b. Certificates – 3rd PARTY REQUESTER		8630
a. All claims confirmed	1846	21.4%
b. All claims canceled	1226	14.2%
c. Claims changed	5558	64.4%
c. Certificates – COMMISSIONER INITIATED REEXAM		174
a. All claims confirmed	17	9.8%
b. All claims canceled	46	26.4%
c. Claims changed	111	63.8%

¹ Total decisions on requests does not include requests that have been vacated or are pending.

² The Director granted requests are only granted petitions on denial under 37 CFR 1.515.

³ Historic certificates were reviewed and the data in items 8-10 reflects the outcome of the review to the extent the data is available as of Oct 1, 2020



United States Patent and Trademark Office

Commissioner for Patents

United States Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2019

1. Total requests filed since start of <i>Ex Parte</i> reexam on 07/01/81.....	13987	
a. By patent owner	3919	28%
b. by other member of the public	9893	71%
c. By order of Commissioner	175	1%
2. Number of Filings by discipline		
a. Chemical Operation	3791	27%
b. Electrical Operation	5214	37%
c. Mechanical Operation	4732	34%
d. Design Patents	250	2%
3. Annual <i>Ex Parte</i> Reexam Filings		
Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No		
1981 78 (3 mos.) 1989 242 1997 380 2005 520 2013 305		
1982 187 1990 297 1998 348 2006 473 2014 355		
1983 186 1991 309 1999 391 2007 600 2015 244		
1984 189 1992 390 2000 312 2008 690 2016 222		
1985 230 1993 359 2001 295 2009 603 2017 191		
1986 232 1994 379 2002 272 2010 697 2018 187		
1987 240 1995 395 2003 394 2011 734 2019 163		
1988 269 1996 413 2004 436 2012 780		
4. Number known to be in litigation.....	1938	14%
5. Decisions on requests ²	14130	
a. No. granted	12905	91%
(1) By examiner	12718	
(2) By Director (on petition)	187	
b. No. denied	1277	9%
(1) By examiner	1225	
(2) Reexam vacated	52	
6. Total examiner denials (includes denials reversed by Director).....	1226	
a. Patent owner requester	524	43%
b. Third party requester	702	57%

1 Total decisions on requests does not include requests that have been vacated or are pending.

2 The Director granted requests are only granted petitions on denial under 37 CFR 1.515.

7. Overall reexamination pendency (Filing date to certificate issue date).....	
a. Average pendency	25.8 (mos.)
b. Median pendency	19.4 (mos.)

8. Total ex parte reexamination certificates issued (1981 - present).....				11980	
	Owner Requester	3rd Party Requester	Comm'r Initiate	Overall	Numbers
a. Certificates with all claims confirmed	6%	15%	0%	20.7%	2480
b. Certificates with all claims canceled	3%	10%	0%	12.7%	1517
c. Certificates with claims changes	19%	46%	1%	66.6%	7983

9. Reexam claim analysis – requester is patent owner or 3rd party or Commissioner initiated.

a. Certificates – PATENT OWNER REQUESTER	3325
a. All claims confirmed	693 21%
b. All claims canceled	300 9%
c. Claims changed	2332 70%
b. Certificates – 3rd PARTY REQUESTER	8481
a. All claims confirmed	1769 21%
b. All claims canceled	1170 14%
c. Claims changed	5542 65%
c. Certificates – COMMISSIONER INITIATED REEXAM	174
a. All claims confirmed	18 10%
b. All claims canceled	47 27%
c. Claims changed	109 63%

1 Total decisions on requests does not include requests that have been vacated or are pending.

2 The Director granted requests are only granted petitions on denial under 37 CFR 1.515.



United States Patent and Trademark Office

Commissioner for Patents

United States Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2018

1. Total requests filed since start of <i>Ex Parte</i> reexam on 07/01/81.....	14005	
a. By patent owner	4137	30%
b. by other member of the public	9825	70%
c. By order of Commissioner	43	0%
2. Number of Filings by discipline		
a. Chemical Operation	3782	27%
b. Electrical Operation	5270	38%
c. Mechanical Operation	4717	34%
d. Design Patents	236	2%
3. Annual <i>Ex Parte</i> Reexam Filings		
Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No		
1981 78 (3 mos.) 1989 242 1997 380 2005 520 2013 305		
1982 187 1990 297 1998 348 2006 473 2014 355		
1983 186 1991 309 1999 391 2007 601 2015 243		
1984 189 1992 390 2000 312 2008 691 2016 225		
1985 230 1993 359 2001 295 2009 605 2017 189		
1986 232 1994 379 2002 272 2010 698 2018 189		
1987 240 1995 395 2003 394 2011 736		
1988 269 1996 413 2004 436 2012 781		
4. Number known to be in litigation.....	4731	34%
5. Decisions on requests.....		13178
a. No. granted	12042	91%
(1) By examiner	11891	
(2) By Director (on petition)	151	
b. No. denied	1136	9%
(1) By examiner	1089	
(2) Reexam vacated	47	
6. Total examiner denials (includes denials reversed by Director).....		2272
a. Patent owner requester	586	26%
b. Third party requester	1686	74%
7. Overall reexamination pendency (Filing date to certificate issue date).....		
a. Average pendency	27.7 (mos.)	
b. Median pendency	19.8 (mos.)	

1 Total decisions on requests does not include requests that have been vacated or are pending.

data as of 9/30/2018

8. Reexam certificate claim analysis:	Owner Requester	3rd Party Requester	Comm'r Initiate	Overall
a. All claims confirmed	6%	15%	0%	21%
b. All claims canceled	3%	10%	0%	12%
c. Claims changed	20%	47%	0%	67%
9. Total ex parte reexamination certificates issued (1981 – present)				11403
a. Certificates with all claims confirmed			2391	21%
b. Certificates with all claims canceled			1404	12%
c. Certificates with claims changes			7608	67%
10. Reexam claim analysis – requester is patent owner or 3rd party or Commissioner initiated.				
a. Certificates – PATENT OWNER REQUESTER				3337
a. All claims confirmed			690	21%
b. All claims canceled			293	9%
c. Claims changed			2313	69%
b. Certificates – 3rd PARTY REQUESTER				8362
a. All claims confirmed			1681	20%
b. All claims canceled			1062	13%
c. Claims changed			5174	62%
c. Certificates – COMMISSIONER INITIATED REEXAM				41
a. All claims confirmed			2	5%
b. All claims canceled			23	56%
c. Claims changed			16	39%



United States Patent and Trademark Office

Commissioner for Patents

United States Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2017

1. Total requests filed since start of <i>Ex Parte</i> reexam on 07/01/81.....	13641	
a. By patent owner	3912	29%
b. by other member of the public	9686	71%
c. By order of Commissioner	43	0%
2. Number of Filings by dicipline		
a. Chemical Operation	3670	27%
b. Electrical Operation	5148	38%
c. Mechanical Operation	4596	34%
d. Design Patents	227	2%
3. Annual <i>Ex Parte</i> Reexam Filings		
Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No		
1981 78 (3 mos.) 1989 242 1997 380 2005 520 2013 305		
1982 187 1990 297 1998 348 2006 473 2014 355		
1983 186 1991 309 1999 391 2007 601 2015 243		
1984 189 1992 390 2000 312 2008 691 2016 219		
1985 230 1993 359 2001 295 2009 605 2017 191		
1986 232 1994 379 2002 272 2010 698		
1987 240 1995 395 2003 394 2011 736		
1988 269 1996 413 2004 436 2012 781		
4. Number known to be in litigation.....	4648	34%
5. Decisions on requests.....		13147
a. No. granted	12047	92%
(1) By examiner	11913	
(2) By Director (on petition)	134	
b. No. denied	1100	8%
(1) By examiner	1054	
(2) Reexam vacated	46	
6. Total examiner denials (includes denials reversed by Director).....		1056
a. Patent owner requester	494	47%
b. Third party requester	562	53%
7. Overall reexamination pendency (Filing date to certificate issue date).....		
a. Average pendency	26.4 (mos.)	
b. Median pendency	19.4 (mos.)	

1 Total decisions on requests does not include requests that have been vacated or are pending.

data as of 9/30/2016

8. Reexam certificate claim analysis:	Owner Requester	3rd Party Requester	Comm'r Initiate	Overall
a. All claims confirmed	6%	15%	0%	21%
b. All calims canceled	3%	9%	0%	12%
c. Claims changed	21%	46%	0%	67%
9. Total ex parte reexamination certificates issued (1981 – present)				11403
a. Certificates with all claims confirmed			2391	21%
b. Certificates with all claims canceled			1404	12%
c. Certificates with claims changes			7608	67%
10. Reexam claim analysis – requester is patent owner or 3rd party or Commissioner initiated.				
a. Certificates – PATENT OWNER REQUESTER				3296
a. All claims confirmed			690	21%
b. All calims canceled			293	9%
c. Claims changed			2313	70%
b. Certificates – 3rd PARTY REQUESTER				7917
a. All claims confirmed			1681	21%
b. All calims canceled			1062	13%
c. Claims changed			5174	65%
c. Certificates – COMMISSIONER INITIATED REEXAM				41
a. All claims confirmed			2	5%
b. All calims canceled			23	56%
c. Claims changed			16	39%



United States Patent and Trademark Office

Commissioner for Patents

United States Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2016

1. Total requests filed since start of <i>Ex Parte</i> reexam on 07/01/81.....	13450	
a. By patent owner	3887	29%
b. by other member of the public	9520	71%
c. By order of Commissioner	43	0%
2. Number of Filings by dicipline		
a. Chemical Operation	3632	27%
b. Electrical Operation	5085	38%
c. Mechanical Operation	4521	34%
d. Design Patents	212	2%
3. Annual <i>Ex Parte</i> Reexam Filings		
Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No		
1981 78 (3 mos.) 1989 242 1997 380 2005 520 2013 305		
1982 187 1990 297 1998 348 2006 473 2014 355		
1983 186 1991 309 1999 391 2007 601 2015 243		
1984 189 1992 390 2000 312 2008 691 2015 219		
1985 230 1993 359 2001 295 2009 605		
1986 232 1994 379 2002 272 2010 698		
1987 240 1995 395 2003 394 2011 736		
1988 269 1996 413 2004 436 2012 781		
4. Number known to be in litigation.....	4350	32%
5. Decisions on requests.....		12929
a. No. granted	11835	92%
(1) By examiner	11704	
(2) By Director (on petition)	131	
b. No. denied	1094	8%
(1) By examiner	1048	
(2) Reexam vacated	46	
6. Total examiner denials (includes denials reversed by Director).....		1048
a. Patent owner requester	492	47%
b. Third party requester	556	53%
7. Overall reexamination pendency (Filing date to certificate issue date).....		
a. Average pendency	25.77 (mos.)	
b. Median pendency	20.2 (mos.)	

1 Total decisions on requests does not include requests that have been vacated or are pending.

data as of 9/30/2016

8. Reexam certificate claim analysis:	Owner Requester	3rd Party Requester	Comm'r Initiate	Overall
a. All claims confirmed	6%	15%	0%	21%
b. All calims canceled	3%	9%	0%	12%
c. Claims changed	21%	45%	0%	66%
9. Total ex parte reexamination certificates issued (1981 – present)				10979
a. Certificates with all claims confirmed			2337	21%
b. Certificates with all claims canceled			1339	12%
c. Certificates with claims changes			7303	67%
10. Reexam claim analysis – requester is patent owner or 3rd party or Commissioner initiated.				
a. Certificates – PATENT OWNER REQUESTER				3270
a. All claims confirmed			685	21%
b. All calims canceled			291	9%
c. Claims changed			2294	70%
b. Certificates – 3rd PARTY REQUESTER				7519
a. All claims confirmed			1631	22%
b. All calims canceled			999	13%
c. Claims changed			4889	65%
c. Certificates – COMMISSIONER INITIATED REEXAM				41
a. All claims confirmed			2	5%
b. All calims canceled			23	56%
c. Claims changed			16	39%

1 Total decisions on requests does not include requests that have been vacated or are pending.



United States Patent and Trademark Office

Commissioner for Patents

United States Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2015

1. Total requests filed since start of <i>Ex Parte</i> reexam on 07/01/81.....	13136	
a. By patent owner	3957	30%
b. by other member of the public	9014	69%
c. By order of Commissioner	165	1%
2. Number of Filings by dicipline		
a. Chemical Operation	3569	27%
b. Electrical Operation	5265	40%
c. Mechanical Operation	4418	34%
d. Design Patents	220	2%
3. Annual <i>Ex Parte</i> Reexam Filings		
Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No		
1981 78 (3 mos.) 1989 243 1997 376 2005 524 2013 296		
1982 187 1990 297 1998 350 2006 511 2014 342		
1983 186 1991 307 1999 385 2007 643 2015 240		
1984 189 1992 392 2000 318 2008 680		
1985 230 1993 359 2001 296 2009 658		
1986 232 1994 379 2002 272 2010 780		
1987 240 1995 392 2003 392 2011 759		
1988 268 1996 418 2004 441 2012 787		
4. Number known to be in litigation.....	4819	37%
5. Decisions on requests.....		12641
a. No. granted	11623	92%
(1) By examiner	11492	
(2) By Director (on petition)	131	
b. No. denied	1018	8%
(1) By examiner	980	
(2) Reexam vacated	38	
6. Total examiner denials (includes denials reversed by Director).....		1118
a. Patent owner requester	496	44%
b. Third party requester	622	56%
7. Overall reexamination pendency (Filing date to certificate issue date).....		
a. Average pendency	26.5 (mos.)	
b. Median pendency	19.6 (mos.)	

1 Total decisions on requests does not include requests that have been vacated or are pending.

data as of 9/30/2015

8. Reexam certificate claim analysis:	Owner Requester	3rd Party Requester	Comm'r Initiate	Overall
a. All claims confirmed	6%	15%	0%	21%
b. All calims canceled	3%	9%	0%	12%
c. Claims changed	21%	45%	1%	67%
9. Total ex parte reexamination certificates issued (1981 – present)				10951
a. Certificates with all claims confirmed			2331	21%
b. Certificates with all claims canceled			1334	12%
c. Certificates with claims changes			7286	67%
10. Reexam claim analysis – requester is patent owner or 3rd party or Commissioner initiated.				
a. Certificates – PATENT OWNER REQUESTER				3268
a. All claims confirmed			685	21%
b. All calims canceled			291	9%
c. Claims changed			2292	70%
b. Certificates – 3rd PARTY REQUESTER				7513
a. All claims confirmed			1628	22%
b. All calims canceled			998	13%
c. Claims changed			4887	65%
c. Certificates – COMMISSIONER INITIATED REEXAM				170
a. All claims confirmed			18	11%
b. All calims canceled			45	26%
c. Claims changed			107	63%



United States Patent and Trademark Office

Commissioner for Patents

United States Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2014

1. Total requests filed since start of <i>Ex Parte</i> reexam on 07/01/81.....	13217	
a. By patent owner	3833	29%
b. by other member of the public	9217	70%
c. By order of Commissioner	167	1%
2. Number of Filings by dicipline		
a. Chemical Operation	3516	27%
b. Electrical Operation	5167	39%
c. Mechanical Operation	4330	33%
d. Design Patents	204	2%
3. Annual <i>Ex Parte</i> Reexam Filings		
Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No		
1981 78 (3 mos.) 1989 243 1997 376 2005 524 2013 305		
1982 187 1990 297 1998 350 2006 511 2014 343		
1983 186 1991 307 1999 385 2007 643		
1984 189 1992 392 2000 318 2008 680		
1985 230 1993 359 2001 296 2009 658		
1986 232 1994 379 2002 272 2010 780		
1987 240 1995 392 2003 392 2011 759		
1988 268 1996 418 2004 441 2012 787		
4. Number known to be in litigation.....	4362	33%
5. Decisions on requests.....		12429
a. No. granted	11402	92%
(1) By examiner	11271	
(2) By Director (on petition)	131	
b. No. denied	1027	8%
(1) By examiner	990	
(2) Reexam vacated	37	
6. Total examiner denials (includes denials reversed by Director).....		1067
a. Patent owner requester	665	62%
b. Third party requester	621	58%
7. Overall reexamination pendency (Filing date to certificate issue date).....		
a. Average pendency	22.3 (mos.)	
b. Median pendency	15.9 (mos.)	

1 Total decisions on requests does not include requests that have been vacated or are pending.

data as of 9/30/2014

8. Reexam certificate claim analysis:	Owner Requester	3rd Party Requester	Comm'r Initiate	Overall
a. All claims confirmed	6%	15%	0%	22%
b. All calims canceled	3%	9%	0%	12%
c. Claims changed	21%	44%	1%	66%
9. Total ex parte reexamination certificates issued (1981 – present)				10698
a. Certificates with all claims confirmed			2303	22%
b. Certificates with all claims canceled			1301	12%
c. Certificates with claims changes			7094	66%
10. Reexam claim analysis – requester is patent owner or 3rd party or Commissioner initiated.				
a. Certificates – PATENT OWNER REQUESTER				3256
a. All claims confirmed			685	21%
b. All calims canceled			289	9%
c. Claims changed			2282	70%
b. Certificates – 3rd PARTY REQUESTER				7272
a. All claims confirmed			1600	22%
b. All calims canceled			967	13%
c. Claims changed			4705	65%
c. Certificates – COMMISSIONER INITIATED REEXAM				170
a. All claims confirmed			18	11%
b. All calims canceled			45	26%
c. Claims changed			107	63%

1 Total decisions on requests does not include requests that have been vacated or are pending.



United States Patent and Trademark Office

Commissioner for Patents

United States Patent and Trademark Office

P.O. Box 1450

Alexandria, VA 22313-1450

www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2013

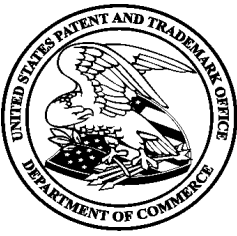
1. Total requests filed since start of <i>Ex Parte</i> reexam on 07/01/81.....	12874	
a. By patent owner	3833	30%
b. by other member of the public	8874	69%
c. By order of Commissioner	167	1%
2. Number of Filings by dicipline		
a. Chemical Operation	3435	27%
b. Electrical Operation	5001	39%
c. Mechanical Operation	4234	33%
d. Design Patents	204	2%
3. Annual <i>Ex Parte</i> Reexam Filings		
Fiscal Yr. No	Fiscal Yr. No	Fiscal Yr. No
1981 78 (3 mos.	1989 243	1997 376
1982 187	1990 297	1998 350
1983 186	1991 307	1999 385
1984 189	1992 392	2000 318
1985 230	1993 359	2001 296
1986 232	1994 379	2002 272
1987 240	1995 392	2003 392
1988 268	1996 418	2004 441
2005 524	2006 511	2007 643
2008 680	2009 658	2010 780
2011 759	2012 787	2013 305
4. Number known to be in litigation.....	4167	32%
5. Decisions on requests.....	12016	
a. No. granted	11013	92%
(1) By examiner	10891	
(2) By Director (on petition)	122	
b. No. denied	1003	8%
(1) By examiner	968	
(2) Reexam vacated	35	
6. Total examiner denials (includes denials reversed by Director).....	1067	
a. Patent owner requester	498	47%
b. Third party requester	606	57%
7. Overall reexamination pendency (Filing date to certificate issue date)		
a. Average pendency	27.8 (mos.)	
b. Median pendency	20.1 (mos.)	

1 Total decisions on requests does not include requests that have been vacated or are pending.

updated as of 11/22/13

8. Reexam certificate claim analysis:	Owner Requeste	3rd Party Requeste	Comm'r Initiate	Overall
a. All claims confirmed	6%	15%	0%	21%
b. All calims canceled	3%	8%	0%	11%
c. Claims changed	21%	40%	1%	62%
9. Total ex parte reexamination certificates issued (1981 – present)				9991
a. Certificates with all claims confirmed			2241	22%
b. Certificates with all claims canceled			1205	12%
c. Certificates with claims changes			6545	66%
10. Reexam claim analysis – requester is patent owner or 3rd party or Commissioner initiated.				
a. Certificates – PATENT OWNER REQUESTER				3279
a. All claims confirmed			682	21%
b. All calims canceled			283	9%
c. Claims changed			2239	68%
b. Certificates – 3rd PARTY REQUESTER				7174
a. All claims confirmed			1541	21%
b. All calims canceled			882	12%
c. Claims changed			4201	59%
c. Certificates – COMMISSIONER INITIATED REEXAM				164
a. All claims confirmed			18	11%
b. All calims canceled			40	24%
c. Claims changed			105	64%

1 Total decisions on requests does not include requests that have been vacated or are pending.



United States Patent and Trademark Office

Commissioner for Patents
 United States Patent and Trademark Office
 P.O. Box 1450
 Alexandria, VA 22313-1450
www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2012

1. Total requests filed since start of <i>ex parte</i> reexam on 07/01/81.....	12569
a. By patent owner	3872 31%
b. by other member of the public	8532 68%
c. By order of Commissioner	165 1%
2. Number a. Chemical Operation	3373 27%
b. Electrical Operation	4831 38%
c. Mechanical Operation	4178 33%
d. Design Patents	187 1%
3. Annual <i>Ex Parte</i> Reexam Filings	
Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No Fiscal Yr. No	
1981 78 (3 mos.) 1989 243 1997 376 2005 524	
1982 187 1990 297 1998 350 2006 511	
1983 186 1991 307 1999 385 2007 643	
1984 189 1992 392 2000 318 2008 680	
1985 230 1993 359 2001 296 2009 658	
1986 232 1994 379 2002 272 2010 780	
1987 240 1995 392 2003 392 2011 759	
1988 268 1996 418 2004 441 2012 788	
4. Number known to be in litigation.....	3994 32%
5. Decisions on requests.....	11737
a. No. granted.....	10755 92%
(1) By examiner	10762
(2) By Director (on petition)	122
b. No. denied.....	982 8%
(1) By examiner	961
(2) Reexam vacated	35
6. Total examiner denials (includes denials reversed by Director).....	1078
a. Patent owner requester	496 46%
b. Third party requester	582 54%
7. Overall reexamination pendency (Filing date to certificate issue date)	
a. Average pendency	27.9 (mos.)
b. Median pendency	20.5 (mos.)

8. Reexam certificate claim analysis:	Owner Requester	3rd Party Requester	Comm'r Initiate	Overall
a. All claims confirmed	21%	22%	11%	21%
b. All calims canceled	8%	12%	24%	11%
c. Claims changed	71%	66%	65%	68%
9. Total ex parte reexamination certificates issued (1981 – present)				9328
a. Certificates with all claims confirmed			2029	22%
b. Certificates with all claims canceled			1063	11%
c. Certificates with claims changes			6236	67%
10. Reexam claim analysis – requester is patent owner or 3rd party or Commissioner initiated.				
a. Certificates – PATENT OWNER REQUESTER				3176
a. All claims confirmed			669	21%
b. All calims canceled			280	9%
c. Claims changed			2227	70%
b. Certificates – 3rd PARTY REQUESTER				6026
a. All claims confirmed			1348	22%
b. All calims canceled			745	12%
c. Claims changed			3933	65%
c. Certificates – COMMISSIONER INITIATED REEXAM				164
a. All claims confirmed			18	11%
b. All calims canceled			38	23%
c. Claims changed			108	66%



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
 United States Patent and Trademark Office
 P.O. Box 1450
 Alexandria, VA 22313-1450
www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2011

1. Total requests filed since start of *ex parte* reexam on 07/01/81.....11782¹
 - a. By patent owner 3801 33%
 - b. By other member of public 7815 66%
 - c. By order of Commissioner 166 1%

2. Number of filings by discipline
 - a. Chemical Operation 3211 27%
 - b. Electrical Operation 4405 37%
 - c. Mechanical Operation 3987 34%
 - d. Design Patents 179 2%

3. Annual *Ex Parte* Reexam Filings

Fiscal Yr.	No.	Fiscal Yr.	No.	Fiscal Yr.	No.	Fiscal Yr.	No.
1981	78 (3 mos.)	1989	243	1997	376	2005	524
1982	187	1990	297	1998	350	2006	511
1983	186	1991	307	1999	385	2007	643
1984	189	1992	392	2000	318	2008	680
1985	230	1993	359	2001	296	2009	658
1986	232	1994	379	2002	272	2010	780
1987	240	1995	392	2003	392	2011	759
1988	268	1996	418	2004	441		

4. Number known to be in litigation.....3894.....33%

5. Decisions on requests.....11262
 - a. No. granted.....10333.....92%
 - (1) By examiner 10213
 - (2) By Director (on petition) 120
 - b. No. denied929.....8%
 - (1) By examiner 894
 - (2) Reexam vacated 35

¹Of the requests received in FY 2011, 22 requests have not yet been accorded a filing date, and preprocessing of 35 requests was terminated for failure to comply with the requirements of 37 CFR 1.510. See Clarification of Filing Date Requirements for *Ex Parte* and *Inter Partes* Reexamination Proceedings, Final Rule, 71 Fed. Reg. 44219 (August 4, 2006).

6.	Total examiner denials (includes denials reversed by Director).....	1014		
a.	Patent owner requester	476	47%	
b.	Third party requester	538	53%	
7.	Overall reexamination pendency (Filing date to certificate issue date)			
a.	Average pendency	25.6 (mos.)		
b.	Median pendency	19.9 (mos.)		
8.	Reexam certificate claim analysis:			
		<u>Owner</u>	<u>3rd Party</u>	<u>Comm'r</u>
		<u>Requester</u>	<u>Requester</u>	<u>Initiated</u>
				<u>Overall</u>
a.	All claims confirmed	21%	24%	11%
b.	All claims cancelled	9%	12%	23%
c.	Claims changes	70%	64%	66%
9.	Total <i>ex parte</i> reexamination certificates issued (1981 – present)			8578
a.	Certificates with all claims confirmed		1943	23%
b.	Certificates with all claims canceled		974	11%
c.	Certificates with claims changes		5661	66%
10.	Reexam claim analysis – requester is patent owner or 3 rd party or Commissioner initiated.			
a.	Certificates – PATENT OWNER REQUESTER			3040
	(1) All claims confirmed		651	21%
	(2) All claims canceled		264	9%
	(3) Claim changes		2125	70%
b.	Certificates – 3 rd PARTY REQUESTER			5380
	(1) All claims confirmed		1274	24%
	(2) All claims canceled		674	12%
	(3) Claim changes		3432	64%
c.	Certificates – COMMISSIONER INITIATED REEXAM			158
	(1) All claims confirmed		18	11%
	(2) All claims canceled		36	23%
	(3) Claim changes		104	66%



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
 United States Patent and Trademark Office
 P.O. Box 1450
 Alexandria, VA 22313-1450
www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2010

1. Total requests filed since start of <i>ex parte</i> reexam on 07/01/81.....	11023 ¹
a. By patent owner	3697 34%
b. By other member of public	7161 65%
c. By order of Commissioner	165 1%
2. Number of filings by discipline	
a. Chemical Operation	3068 28%
b. Electrical Operation	4010 36%
c. Mechanical Operation	3771 34%
d. Design Patents	174 2%
3. Annual <i>Ex Parte</i> Reexam Filings	
Fiscal Yr. No. Fiscal Yr. No. Fiscal Yr. No. Fiscal Yr. No.	
1981 78 (3 mos.) 1989 243 1997 376 2005 524	
1982 187 1990 297 1998 350 2006 511	
1983 186 1991 307 1999 385 2007 643	
1984 189 1992 392 2000 318 2008 680	
1985 230 1993 359 2001 296 2009 658	
1986 232 1994 379 2002 272 2010 780	
1987 240 1995 392 2003 392	
1988 268 1996 418 2004 441	
4. Number known to be in litigation.....	3568.....32%
5. Decisions on requests.....	10495
a. No. granted.....	9648.....92%
(1) By examiner	9534
(2) By Director (on petition)	114
b. No. denied	847.....8%
(1) By examiner	812
(2) Reexam vacated	35

¹Of the requests received in FY 2010, 16 requests have not yet been accorded a filing date, and preprocessing of 80 requests was terminated for failure to comply with the requirements of 37 CFR 1.510. See Clarification of Filing Date Requirements for *Ex Parte* and *Inter Partes* Reexamination Proceedings, Final Rule, 71 Fed. Reg. 44219 (August 4, 2006).

6.	Total examiner denials (includes denials reversed by Director).....	926		
a.	Patent owner requester	451	49%	
b.	Third party requester	475	51%	
7.	Overall reexamination pendency (Filing date to certificate issue date)			
a.	Average pendency	25.5 (mos.)		
b.	Median pendency	20.0 (mos.)		
8.	Reexam certificate claim analysis:			
		<u>Owner</u>	<u>3rd Party</u>	<u>Comm'r</u>
		<u>Requester</u>	<u>Requester</u>	<u>Initiated</u>
				<u>Overall</u>
a.	All claims confirmed	22%	25%	12%
b.	All claims cancelled	8%	13%	23%
c.	Claims changes	70%	62%	65%
9.	Total <i>ex parte</i> reexamination certificates issued (1981 – present)	7782		
a.	Certificates with all claims confirmed	1817	23%	
b.	Certificates with all claims canceled	893	12%	
c.	Certificates with claims changes	5072	65%	
10.	Reexam claim analysis – requester is patent owner or 3 rd party or Commissioner initiated.			
a.	Certificates – PATENT OWNER REQUESTER	2947		
	(1) All claims confirmed	638	22%	
	(2) All claims canceled	251	8%	
	(3) Claim changes	2058	70%	
b.	Certificates – 3 rd PARTY REQUESTER	4681		
	(1) All claims confirmed	1161	25%	
	(2) All claims canceled	607	13%	
	(3) Claim changes	2913	62%	
c.	Certificates – COMMISSIONER INITIATED REEXAM	154		
	(1) All claims confirmed	18	12%	
	(2) All claims canceled	35	23%	
	(3) Claim changes	101	65%	



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2009

1. Total requests filed since start of *ex parte* reexam on 07/01/81..... 10243¹
 - a. By patent owner 3634 35%
 - b. By other member of public 6444 63%
 - c. By order of Commissioner 165 2%

2. Number of filings by discipline
 - a. Chemical Operation 2931 29%
 - b. Electrical Operation 3596 35%
 - c. Mechanical Operation 3554 34%
 - d. Design Patents 162 2%

3. Annual *Ex Parte* Reexam Filings

Fiscal Yr.	No.	Fiscal Yr.	No.	Fiscal Yr.	No.	Fiscal Yr.	No.
1981	78 (3 mos.)	1989	243	1997	376	2005	524
1982	187	1990	297	1998	350	2006	511
1983	186	1991	307	1999	385	2007	643
1984	189	1992	392	2000	318	2008	680
1985	230	1993	359	2001	296	2009	658
1986	232	1994	379	2002	272		
1987	240	1995	392	2003	392		
1988	268	1996	418	2004	441		

4. Number known to be in litigation.....3221.....31%

5. Decisions on requests.....9833
 - a. No. granted.....9041.....92%
 - (1) By examiner 8928
 - (2) By Director (on petition) 113
 - b. No. denied792.....8%
 - (1) By examiner 757
 - (2) Reexam vacated 35

¹Of the requests received in FY 2009, 22 requests have not yet been accorded a filing date, and preprocessing of 41 requests was terminated for failure to comply with the requirements of 37 CFR 1.510. See Clarification of Filing Date Requirements for *Ex Parte* and *Inter Partes* Reexamination Proceedings, Final Rule, 71 Fed. Reg. 44219 (August 4, 2006).

6.	Total examiner denials (includes denials reversed by Director).....	870		
a.	Patent owner requester	445	51%	
b.	Third party requester	425	49%	
7.	Overall reexamination pendency (Filing date to certificate issue date)			
a.	Average pendency	25.2 (mos.)		
b.	Median pendency	19.7 (mos.)		
8.	Reexam certificate claim analysis:			
		<u>Owner</u>	<u>3rd Party</u>	<u>Comm'r</u>
		<u>Requester</u>	<u>Requester</u>	<u>Initiated</u>
				<u>Overall</u>
a.	All claims confirmed	22%	26%	12%
b.	All claims cancelled	8%	13%	23%
c.	Claims changes	70%	61%	65%
9.	Total <i>ex parte</i> reexamination certificates issued (1981 – present)	7089		
a.	Certificates with all claims confirmed	1725	25%	
b.	Certificates with all claims canceled	807	11%	
c.	Certificates with claims changes	4557	64%	
10.	Reexam claim analysis – requester is patent owner or 3 rd party or Commissioner initiated.			
a.	Certificates – PATENT OWNER REQUESTER	2827		
	(1) All claims confirmed	625	22%	
	(2) All claims canceled	233	8%	
	(3) Claim changes	1969	70%	
b.	Certificates – 3 rd PARTY REQUESTER	4111		
	(1) All claims confirmed	1082	26%	
	(2) All claims canceled	540	13%	
	(3) Claim changes	2489	61%	
c.	Certificates – COMMISSIONER INITIATED REEXAM	151		
	(1) All claims confirmed	18	12%	
	(2) All claims canceled	34	23%	
	(3) Claim changes	99	65%	



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
 United States Patent and Trademark Office
 P.O. Box 1450
 Alexandria, VA 22313-1450
www.uspto.gov

Ex Parte Reexamination Filing Data - September 30, 2008

1.	Total requests filed since start of <i>ex partes</i> reexam on 07/01/81	9585 ¹																																																																									
	a. By patent owner	3567	37%																																																																								
	b. By other member of public	5853	61%																																																																								
	c. By order of Commissioner	165	2%																																																																								
2.	Number of filings by discipline																																																																										
	a. Chemical Operation	2811	29%																																																																								
	b. Electrical Operation	3261	34%																																																																								
	c. Mechanical Operation	3357	35%																																																																								
	d. Design Patents	156	2%																																																																								
3.	Annual Ex Parte Reexam Filings																																																																										
	<table><tr><td>Fiscal Yr.</td><td>No.</td><td>Fiscal Yr.</td><td>No.</td><td>Fiscal Yr.</td><td>No.</td><td>Fiscal Yr.</td><td>No.</td></tr><tr><td>1981</td><td>78 (3 mos.)</td><td>1989</td><td>243</td><td>1997</td><td>376</td><td>2005</td><td>524</td></tr><tr><td>1982</td><td>187</td><td>1990</td><td>297</td><td>1998</td><td>350</td><td>2006</td><td>511</td></tr><tr><td>1983</td><td>186</td><td>1991</td><td>307</td><td>1999</td><td>385</td><td>2007</td><td>643</td></tr><tr><td>1984</td><td>189</td><td>1992</td><td>392</td><td>2000</td><td>318</td><td>2008</td><td>680</td></tr><tr><td>1985</td><td>230</td><td>1993</td><td>359</td><td>2001</td><td>296</td><td></td><td></td></tr><tr><td>1986</td><td>232</td><td>1994</td><td>379</td><td>2002</td><td>272</td><td></td><td></td></tr><tr><td>1987</td><td>240</td><td>1995</td><td>392</td><td>2003</td><td>392</td><td></td><td></td></tr><tr><td>1988</td><td>268</td><td>1996</td><td>418</td><td>2004</td><td>441</td><td></td><td></td></tr></table>	Fiscal Yr.	No.	Fiscal Yr.	No.	Fiscal Yr.	No.	Fiscal Yr.	No.	1981	78 (3 mos.)	1989	243	1997	376	2005	524	1982	187	1990	297	1998	350	2006	511	1983	186	1991	307	1999	385	2007	643	1984	189	1992	392	2000	318	2008	680	1985	230	1993	359	2001	296			1986	232	1994	379	2002	272			1987	240	1995	392	2003	392			1988	268	1996	418	2004	441				
Fiscal Yr.	No.	Fiscal Yr.	No.	Fiscal Yr.	No.	Fiscal Yr.	No.																																																																				
1981	78 (3 mos.)	1989	243	1997	376	2005	524																																																																				
1982	187	1990	297	1998	350	2006	511																																																																				
1983	186	1991	307	1999	385	2007	643																																																																				
1984	189	1992	392	2000	318	2008	680																																																																				
1985	230	1993	359	2001	296																																																																						
1986	232	1994	379	2002	272																																																																						
1987	240	1995	392	2003	392																																																																						
1988	268	1996	418	2004	441																																																																						
4.	Number known to be in litigation.....	2849.....	30%																																																																								
5.	Decisions on requests.....	9219																																																																									
	a. No. granted.....	8467.....	92%																																																																								
	(1) By examiner	8354																																																																									
	(2) By Director (on petition)	113																																																																									
	b. No. denied	752.....	8%																																																																								
	(1) By examiner	717																																																																									
	(2) Reexam vacated	35																																																																									

¹Of the requests received in FY 2008, 7 requests have not yet been accorded a filing date, and preprocessing of 15 requests was terminated for failure to comply with the requirements of 37 CFR 1.510. See Clarification of Filing Date Requirements for *Ex Parte* and *Inter Partes* Reexamination Proceedings, Final Rule, 71 Fed. Reg. 44219 (August 4, 2006).

6.	Total examiner denials (includes denials reserved by Director).....	830		
a.	Patent owner requester	441	53%	
b.	Third party requester	389	47%	
7.	Overall reexamination pendency (Filing date to certificate issue date)			
a.	Average pendency	24.5 (mos.)		
b.	Median pendency	19.0 (mos.)		
8.	Reexam certificate claim analysis:			
		<u>Owner</u>	<u>3rd Party</u>	<u>Comm'r</u>
		<u>Requester</u>	<u>Requester</u>	<u>Initiated</u>
				<u>Overall</u>
a.	All claims confirmed	22%	28%	12%
b.	All claims cancelled	8%	13%	21%
c.	Claims changes	70%	59%	67%
9.	Total ex parte reexamination certificates issued (1981 – present)	6457		
a.	Certificates with all claims confirmed	1624	25%	
b.	Certificates with all claims canceled	721	11%	
c.	Certificates with claims changes	4112	64%	
10.	Reexam claim analysis – requester is patent owner or 3 rd party; or Comm'r initiated.			
a.	Certificates – PATENT OWNER REQUESTER	2722		
	(1) All claims confirmed	611	22%	
	(2) All claims canceled	214	8%	
	(3) Claim changes	1897	70%	
b.	Certificates – 3 rd PARTY REQUESTER	3588		
	(1) All claims confirmed	995	28%	
	(2) All claims canceled	476	13%	
	(3) Claim changes	2117	59%	
c.	Certificates – COMM'R INITIATED REEXAM	147		
	(1) All claims confirmed	18	12%	
	(2) All claims canceled	31	21%	
	(3) Claim changes	98	67%	

EXHIBIT 21



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/694,802	09/03/2017	Robert Paul Morris	PMOR0120B	5456

92045	7590	11/29/2017
-------	------	------------

The Caldwell Firm, LLC
 PO Box 59655
 Dept. SVIPGP
 Dallas, TX 75229

EXAMINER	
MEKY, MOUSTAFA M	

ART UNIT	PAPER NUMBER
2457	

NOTIFICATION DATE	DELIVERY MODE
11/29/2017	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

pcaldwell@thecaldwellfirm.com
 eofficeaction@apcoll.com

Office Action Summary**Application No.**
15/694,802**Applicant(s)**
MORRIS, ROBERT PAUL**Examiner**
MOUSTAFA M. MEKY**Art Unit**
2457**AIA (First Inventor to File)
Status**
No**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --****Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09/03/2017.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
- 4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

- 5) ☒ Claim(s) 1-30 is/are pending in the application.
5a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 6) ☐ Claim(s) _____ is/are allowed.
- 7) ☒ Claim(s) 1-30 is/are rejected.
- 8) ☐ Claim(s) _____ is/are objected to.
- 9) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

- 10) ☐ The specification is objected to by the Examiner.
- 11) ☒ The drawing(s) filed on 09/03/2017 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

- a) ☐ All b) ☐ Some** c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)
Paper No(s)/Mail Date _____
- 3) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 4) ☐ Other: _____

Application/Control Number: 15/694,802

Page 2

Art Unit: 2457

1. The present application is being examined under the pre-AIA first to invent provisions.
2. Claims 1-30 are presenting for examination.
3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to

Application/Control Number: 15/694,802

Page 3

Art Unit: 2457

be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1-30 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-22 of U.S. Patent No. 8,219,606.

Although the conflicting claims are not identical, they are not patentably distinct from each other because the patent 606 teaches substantially the claimed limitations.

5. Claims 1-30 would be allow upon receiving the terminal disclaimer.

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MOUSTAFA M. MEKY whose telephone number is (571)272-4005. The examiner can normally be reached on flex.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an

Application/Control Number: 15/694,802

Page 4

Art Unit: 2457

interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on 571-272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MOUSTAFA M MEKY
Primary Examiner
Art Unit 2457

/MOUSTAFA M MEKY/
Primary Examiner, Art Unit 2457

11/24/2017

Doc Code: DIST.E.FILE**Document Description: Electronic Terminal Disclaimer - Filed**U.S. Patent and Trademark Office
Department of Commerce

Electronic Petition Request	TERMINAL DISCLAIMER TO OBIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT
Application Number	15694802
Filing Date	03-Sep-2017
First Named Inventor	Robert Morris
Attorney Docket Number	PMOR0120B
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION

☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action

☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.

Owner	Percent Interest
Sitting Man, LLC	100%

The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)

8219606

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:

- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

☒ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.

- ☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.

Applicant claims the following fee status:

- ☒ Small Entity
- ☐ Micro Entity
- ☐ Regular Undiscounted

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

- ☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application
- Registration Number 44580
- ☐ A sole inventor
- ☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application
- ☐ A joint inventor; all of whom are signing this request

Signature	/Patrick Caldwell/
Name	Patrick E. Caldwell

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

EXHIBIT 22



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.				
15/694,803	09/03/2017	Robert Paul Morris	PMOR0120C	4380				
92045	7590	11/29/2017	<table border="1"> <tr> <td colspan="2">EXAMINER</td> </tr> <tr> <td colspan="2">MEKY, MOUSTAFA M</td> </tr> </table>		EXAMINER		MEKY, MOUSTAFA M	
EXAMINER								
MEKY, MOUSTAFA M								
The Caldwell Firm, LLC PO Box 59655 Dept. SVIPGP Dallas, TX 75229			<table border="1"> <tr> <td>ART UNIT</td> <td>PAPER NUMBER</td> </tr> <tr> <td>2457</td> <td></td> </tr> </table>		ART UNIT	PAPER NUMBER	2457	
ART UNIT	PAPER NUMBER							
2457								
			<table border="1"> <tr> <td>NOTIFICATION DATE</td> <td>DELIVERY MODE</td> </tr> <tr> <td>11/29/2017</td> <td>ELECTRONIC</td> </tr> </table>		NOTIFICATION DATE	DELIVERY MODE	11/29/2017	ELECTRONIC
NOTIFICATION DATE	DELIVERY MODE							
11/29/2017	ELECTRONIC							

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

pcaldwell@thecaldwellfirm.com
 eofficeaction@apcoll.com

Office Action Summary**Application No.**
15/694,803**Applicant(s)**
MORRIS, ROBERT PAUL**Examiner**
MOUSTAFA M. MEKY**Art Unit**
2457**AIA (First Inventor to File)
Status**
No**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --****Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09/03/2017.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
- 4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

- 5) ☒ Claim(s) 1-30 is/are pending in the application.
 5a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 6) ☐ Claim(s) _____ is/are allowed.
- 7) ☒ Claim(s) 1-30 is/are rejected.
- 8) ☐ Claim(s) _____ is/are objected to.
- 9) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

- 10) ☐ The specification is objected to by the Examiner.
- 11) ☒ The drawing(s) filed on 09/03/2017 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

- a) ☐ All b) ☐ Some** c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)
 Paper No(s)/Mail Date _____
- 3) ☐ Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____
- 4) ☐ Other: _____

Application/Control Number: 15/694,803

Page 2

Art Unit: 2457

1. The present application is being examined under the pre-AIA first to invent provisions.
2. Claims 1-30 are presenting for examination.
3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to

Application/Control Number: 15/694,803

Page 3

Art Unit: 2457

be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1-30 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-22 of U.S. Patent No. 8,219,606.

Although the conflicting claims are not identical, they are not patentably distinct from each other because the patent 606 teaches substantially the claimed limitations.

5. Claims 1-30 would be allow upon receiving the terminal disclaimer.

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MOUSTAFA M. MEKY whose telephone number is (571)272-4005. The examiner can normally be reached on flex.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview,

Application/Control Number: 15/694,803

Page 4

Art Unit: 2457

applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on 571-272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/MOUSTAFA M MEKY/

Primary Examiner, Art Unit 2457

11/24/2017

Doc Code: DIST.E.FILE**Document Description: Electronic Terminal Disclaimer - Filed**U.S. Patent and Trademark Office
Department of Commerce

Electronic Petition Request	TERMINAL DISCLAIMER TO OBIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT
Application Number	15694803
Filing Date	03-Sep-2017
First Named Inventor	Robert Morris
Attorney Docket Number	PMOR0120C
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION

☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action

☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.

Owner	Percent Interest
Sitting Man, LLC	100%

The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)

8219606

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:

- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

☒ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.

- ☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.

Applicant claims the following fee status:

- ☒ Small Entity
- ☐ Micro Entity
- ☐ Regular Undiscounted

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

- ☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application
- Registration Number 44580
- ☐ A sole inventor
- ☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application
- ☐ A joint inventor; all of whom are signing this request

Signature	/Patrick Caldwell/
Name	Patrick E. Caldwell

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

EXHIBIT 23



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/915,053	03/07/2018	Robert Paul Morris	PMOR0120F	3111

92045	7590	06/07/2018
The Caldwell Firm, LLC		
PO Box 59655		
Dept. SVIPGP		
Dallas, TX 75229		

EXAMINER	
MEKY, MOUSTAFA M	

ART UNIT	PAPER NUMBER
2457	

NOTIFICATION DATE	DELIVERY MODE
06/07/2018	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

pcaldwell@thecaldwellfirm.com
 eofficeaction@apcoll.com

Office Action Summary**Application No.**
15/915,053**Applicant(s)**
MORRIS, ROBERT PAUL**Examiner**
MOUSTAFA M. MEKY**Art Unit**
2457**AIA (First Inventor to File)
Status**
No**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --****Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 03/07/2018.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
- 4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

- 5) ☒ Claim(s) 1-30 is/are pending in the application.
5a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 6) ☐ Claim(s) _____ is/are allowed.
- 7) ☒ Claim(s) 1-30 is/are rejected.
- 8) ☐ Claim(s) _____ is/are objected to.
- 9) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

- 10) ☐ The specification is objected to by the Examiner.
- 11) ☒ The drawing(s) filed on 03/07/2018 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

- a) ☐ All b) ☐ Some** c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)
Paper No(s)/Mail Date 03/07/2018
- 3) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 4) ☐ Other: _____

Application/Control Number: 15/915,053

Page 2

Art Unit: 2457

1. The present application is being examined under the pre-AIA first to invent provisions.

Detailed Action

Double patenting Rejection

2. Claims 1-30 are presenting for examination.
3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory

Application/Control Number: 15/915,053

Page 3

Art Unit: 2457

double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1-30 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-30 of U.S. Patent No. 9,923,996.

Although the conflicting claims are not identical, they are not patentably distinct from each other because the patent 996 teach substantially the claimed limitations.

5. Claims 1-30 would be allow upon receiving the terminal disclaimer.

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to MOUSTAFA M. MEKY whose telephone number is (571)272-4005. The examiner can normally be reached on flex.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview,

Application/Control Number: 15/915,053

Page 4

Art Unit: 2457

applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on 571-272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/MOUSTAFA M MEKY/

Primary Examiner, Art Unit 2457

05/29/2018

Doc Code: DIST.E.FILE**Document Description: Electronic Terminal Disclaimer - Filed**U.S. Patent and Trademark Office
Department of Commerce

Electronic Petition Request	TERMINAL DISCLAIMER TO OBIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT
Application Number	15915053
Filing Date	07-Mar-2018
First Named Inventor	Robert Morris
Attorney Docket Number	PMOR0120F
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION

☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action

☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.

Owner	Percent Interest
Sitting Man, LLC	100%

The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)

9923996

9923995

8219606

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:

- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

- ☒ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.
- ☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.

Applicant claims the following fee status:

- ☒ Small Entity
- ☐ Micro Entity
- ☐ Regular Undiscounted

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

- ☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application
- Registration Number 44580
- ☐ A sole inventor
- ☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application
- ☐ A joint inventor; all of whom are signing this request

Signature	/Patrick Caldwell/
Name	Patrick Caldwell

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

EXHIBIT 24



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/915,047	03/07/2018	Robert Paul Morris	PMOR0120D	8628

92045	7590	05/31/2018
-------	------	------------

The Caldwell Firm, LLC
 PO Box 59655
 Dept. SVIPGP
 Dallas, TX 75229

EXAMINER	
MEKY, MOUSTAFA M	

ART UNIT	PAPER NUMBER
2457	

NOTIFICATION DATE	DELIVERY MODE
05/31/2018	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

pcaldwell@thecaldwellfirm.com
 eofficeaction@apcoll.com

Office Action Summary**Application No.**
15/915,047**Applicant(s)**
MORRIS, ROBERT PAUL**Examiner**
MOUSTAFA M. MEKY**Art Unit**
2457**AIA (First Inventor to File)
Status**
No**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --****Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 03/07/2018.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
- 4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

- 5) ☒ Claim(s) 1-30 is/are pending in the application.
 5a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 6) ☐ Claim(s) _____ is/are allowed.
- 7) ☒ Claim(s) 1-30 is/are rejected.
- 8) ☐ Claim(s) _____ is/are objected to.
- 9) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

- 10) ☐ The specification is objected to by the Examiner.
- 11) ☒ The drawing(s) filed on 03/07/2018 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

- a) ☐ All b) ☐ Some** c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)
 Paper No(s)/Mail Date 03/07/2018
- 3) ☐ Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____
- 4) ☐ Other: _____

Application/Control Number: 15/915,047

Page 2

Art Unit: 2457

1. The present application is being examined under the pre-AIA first to invent provisions.

Detailed Action

Double patenting Rejection

2. Claims 1-30 are presenting for examination.
3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory

Application/Control Number: 15/915,047

Page 3

Art Unit: 2457

double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1-30 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-22 of U.S. Patent No. 8,219,606 & claims 1-30 of U.S. Patent No. 9,923,995. Although the conflicting claims are not identical, they are not patentably distinct from each other because the patents 606 & 995 teach substantially the claimed limitations.

5. Claims 1-30 would be allow upon receiving the terminal disclaimer.

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to MOUSTAFA M. MEKY whose telephone number is (571)272-4005. The examiner can normally be reached on flex.

Application/Control Number: 15/915,047

Page 4

Art Unit: 2457

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on 571-272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/MOUSTAFA M MEKY/

Primary Examiner, Art Unit 2457

05/24/2018

Doc Code: DIST.E.FILE**Document Description: Electronic Terminal Disclaimer - Filed**U.S. Patent and Trademark Office
Department of Commerce

Electronic Petition Request	TERMINAL DISCLAIMER TO OBIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT
Application Number	15915047
Filing Date	07-Mar-2018
First Named Inventor	Robert Morris
Attorney Docket Number	PMOR0120D
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION

☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action

☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.

Owner	Percent Interest
Sitting Man, LLC	100%

The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)

9923996

9923995

8219606

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:

- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

- ☒ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.
- ☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.

Applicant claims the following fee status:

- ☒ Small Entity
- ☐ Micro Entity
- ☐ Regular Undiscounted

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

- ☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application
- Registration Number 44580
- ☐ A sole inventor
- ☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application
- ☐ A joint inventor; all of whom are signing this request

Signature	/Patrick Caldwell/
Name	Patrick Caldwell

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

EXHIBIT 25



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/915,052	03/07/2018	Robert Paul Morris	PMOR0120E	3251

92045	7590	06/05/2018
-------	------	------------

The Caldwell Firm, LLC
 PO Box 59655
 Dept. SVIPGP
 Dallas, TX 75229

EXAMINER	
MEKY, MOUSTAFA M	

ART UNIT	PAPER NUMBER
2457	

NOTIFICATION DATE	DELIVERY MODE
06/05/2018	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

pcaldwell@thecaldwellfirm.com
 eofficeaction@apcoll.com

Office Action Summary**Application No.**
15/915,052**Applicant(s)**
MORRIS, ROBERT PAUL**Examiner**
MOUSTAFA M. MEKY**Art Unit**
2457**AIA (First Inventor to File)
Status**
Yes**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --****Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 03/07/2018.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.
- 4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

- 5) ☒ Claim(s) 1-30 is/are pending in the application.
 5a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 6) ☐ Claim(s) _____ is/are allowed.
- 7) ☒ Claim(s) 1-30 is/are rejected.
- 8) ☐ Claim(s) _____ is/are objected to.
- 9) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

- 10) ☐ The specification is objected to by the Examiner.
- 11) ☒ The drawing(s) filed on 03/07/2018 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

- a) ☐ All b) ☐ Some** c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)
 Paper No(s)/Mail Date 03/07/2018
- 3) ☐ Interview Summary (PTO-413)
 Paper No(s)/Mail Date. _____
- 4) ☐ Other: _____

Application/Control Number: 15/915,052

Page 2

Art Unit: 2457

1. The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

Detailed Action

Double patenting Rejection

2. Claims 1-30 are presenting for examination.

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

Application/Control Number: 15/915,052

Page 3

Art Unit: 2457

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1-30 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-30 of U.S. Patent No. 9,923,996.

Although the conflicting claims are not identical, they are not patentably distinct from each other because the patent 996 teach substantially the claimed limitations.

5. Claims 1-30 would be allow upon receiving the terminal disclaimer.

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to MOUSTAFA M. MEKY whose telephone number is (571)272-4005. The examiner can normally be reached on flex.

Application/Control Number: 15/915,052

Page 4

Art Unit: 2457

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on 571-272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/MOUSTAFA M MEKY/

Primary Examiner, Art Unit 2457

05/29/2018

Doc Code: DIST.E.FILE**Document Description: Electronic Terminal Disclaimer - Filed**U.S. Patent and Trademark Office
Department of Commerce

Electronic Petition Request	TERMINAL DISCLAIMER TO OBIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT
Application Number	15915052
Filing Date	07-Mar-2018
First Named Inventor	Robert Morris
Attorney Docket Number	PMOR0120E
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION

☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action

☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.

Owner	Percent Interest
Sitting Man, LLC	100%

The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)

9923996

9923995

8219606

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:

- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

- ☒ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.
- ☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.

Applicant claims the following fee status:

- ☒ Small Entity
- ☐ Micro Entity
- ☐ Regular Undiscounted

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

- ☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application
- Registration Number 44580
- ☐ A sole inventor
- ☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application
- ☐ A joint inventor; all of whom are signing this request

Signature	/Patrick Caldwell/
Name	Patrick Caldwell

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

EXHIBIT 26



UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/040,517	07/19/2018	Robert Paul Morris	PMOR0120G	1165
92045	7590	10/04/2018		
The Caldwell Firm, LLC PO Box 59655 Dept. SVIPGP Dallas, TX 75229			EXAMINER MEKY, MOUSTAFA M	
			ART UNIT	PAPER NUMBER
			2457	
			NOTIFICATION DATE	DELIVERY MODE
			10/04/2018	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eofficeaction@appcoll.com
pcaldwell@thecaldwellfirm.com

Office Action Summary**Application No.**

16/040,517

Applicant(s)

Morris, Robert Paul

Examiner

Moustafa M Meky

Art Unit

2457

AIA Status

No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on 07/19/2018.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on ____.

2a) ☐ This action is **FINAL**.

2b) ☒ This action is non-final.

3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.

4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5) ☒ Claim(s) 1-20 is/are pending in the application.

5a) Of the above claim(s) ____ is/are withdrawn from consideration.

6) ☐ Claim(s) ____ is/are allowed.

7) ☒ Claim(s) 1-20 is/are rejected.

8) ☐ Claim(s) ____ is/are objected to.

9) ☐ Claim(s) ____ are subject to restriction and/or election requirement

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10) ☐ The specification is objected to by the Examiner.

11) ☒ The drawing(s) filed on See Continuation Sheet is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a) ☐ All b) ☐ Some** c) ☐ None of the:

1. ☐ Certified copies of the priority documents have been received.

2. ☐ Certified copies of the priority documents have been received in Application No. ____.

3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

3) ☐ Interview Summary (PTO-413)

Paper No(s)/Mail Date ____.

2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)

4) ☐ Other: ____.

Paper No(s)/Mail Date 07/19/2018.

Continuation Sheet (PTOL-326)

Application No. 16/040,517

Continuation of Application Papers 11): 07/19/2018

Application/Control Number: 16/040,517
Art Unit: 2457

Page 2

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

1. Claims 1-20 are presenting for examination.

Detailed Action

Statutory Double Patenting Rejection

2. A rejection based on double patenting of the “same invention” type finds its support in the language of 35 U.S.C. 101 which states that “whoever invents or discovers any new and useful process... may obtain a patent therefor...” (Emphasis added). Thus, the term “same invention,” in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the claims that are directed to the same invention so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

3. Claims 1-20 are rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 1-14 & 25-30 of prior U.S. Patent No. 9,923,995. This is a statutory double patenting rejection.

Application/Control Number: 16/040,517
Art Unit: 2457

Page 3

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Moustafa M Meky whose telephone number is (571)272-4005. The examiner can normally be reached on Monday-Friday 9AM-5PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on 571-272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MOUSTAFA M. MEKY
Primary Patent Examiner
Art Unit 2457

/MOUSTAFA M MEKY/
Primary Examiner, Art Unit 2457

09/30/2018

Doc Code: DIST.E.FILE**Document Description: Electronic Terminal Disclaimer - Filed**U.S. Patent and Trademark Office
Department of Commerce

Electronic Petition Request	TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT	
Application Number	16040517	
Filing Date	19-Jul-2018	
First Named Inventor	Robert Morris	
Attorney Docket Number	PMOR0120G	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION	
<input checked="" type="checkbox"/> Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action <input checked="" type="checkbox"/> This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.		
Owner	Percent Interest	
Jenam Tech, LLC	100%	
The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s) 9923996 10075564 10075565 10069945 8219606 9923995		

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:

- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

☒ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.

☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.

Applicant claims the following fee status:

- ☒ Small Entity
- ☐ Micro Entity
- ☐ Regular Undiscounted

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application

Registration Number 44580

☐ A sole inventor

☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application

☐ A joint inventor; all of whom are signing this request

Signature	/Patrick Caldwell/
Name	Patrick Caldwell

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

Electronic Patent Application Fee Transmittal

Application Number:	16040517			
Filing Date:	19-Jul-2018			
Title of Invention:	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION			
First Named Inventor/Applicant Name:	Robert Paul Morris			
Filer:	Patrick Edgar Caldwell			
Attorney Docket Number:	PMOR0120G			
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
STATUTORY OR TERMINAL DISCLAIMER	2814	1	160	160
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

EXHIBIT 27



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/040,522	07/19/2018	Robert Paul Morris	PMOR0120H	8426
92045	7590	10/04/2018		
The Caldwell Firm, LLC PO Box 59655 Dept. SVIPGP Dallas, TX 75229			EXAMINER MEKY, MOUSTAFA M	
			ART UNIT	PAPER NUMBER
			2457	
			NOTIFICATION DATE	DELIVERY MODE
			10/04/2018	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eofficeaction@appcoll.com
pcaldwell@thecaldwellfirm.com

Office Action Summary**Application No.**

16/040,522

Applicant(s)

Morris, Robert Paul

Examiner

Moustafa M Meky

Art Unit

2457

AIA Status

No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on 07/19/2018.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on ____.

2a) ☐ This action is **FINAL**.

2b) ☒ This action is non-final.

3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.

4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5) ☒ Claim(s) 1-20 is/are pending in the application.

5a) Of the above claim(s) ____ is/are withdrawn from consideration.

6) ☐ Claim(s) ____ is/are allowed.

7) ☒ Claim(s) 1-20 is/are rejected.

8) ☐ Claim(s) ____ is/are objected to.

9) ☐ Claim(s) ____ are subject to restriction and/or election requirement

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10) ☐ The specification is objected to by the Examiner.

11) ☒ The drawing(s) filed on See Continuation Sheet is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a) ☐ All b) ☐ Some** c) ☐ None of the:

1. ☐ Certified copies of the priority documents have been received.

2. ☐ Certified copies of the priority documents have been received in Application No. ____.

3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

3) ☐ Interview Summary (PTO-413)

Paper No(s)/Mail Date ____.

2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)

4) ☐ Other: ____.

Paper No(s)/Mail Date 07/19/2018.

Continuation Sheet (PTOL-326)

Application No. 16/040,522

Continuation of Application Papers 11): 07/19/2018

Application/Control Number: 16/040,522
Art Unit: 2457

Page 2

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

1. Claims 1-20 are presenting for examination.

Detailed Action

Statutory Type Double Patenting Rejection

2. A rejection based on double patenting of the “same invention” type finds its support in the language of 35 U.S.C. 101 which states that “whoever invents or discovers any new and useful process... may obtain a patent therefor...” (Emphasis added). Thus, the term “same invention,” in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the claims that are directed to the same invention so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

3. Claims 1-20 are rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 1-14 & 25-30 of prior U.S. Patent No. 9,923,995. This is a statutory double patenting rejection.

Application/Control Number: 16/040,522
Art Unit: 2457

Page 3

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Moustafa M Meky whose telephone number is (571)272-4005. The examiner can normally be reached on Monday-Friday 9AM-5PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on 571-272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MOUSTAFA M. MEKY
Primary Patent Examiner
Art Unit 2457

/MOUSTAFA M MEKY/
Primary Examiner, Art Unit 2457

09/30/2018

Doc Code: DIST.E.FILE**Document Description: Electronic Terminal Disclaimer - Filed**U.S. Patent and Trademark Office
Department of Commerce

Electronic Petition Request	TERMINAL DISCLAIMER TO OBIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT
Application Number	16040522
Filing Date	19-Jul-2018
First Named Inventor	Robert Morris
Attorney Docket Number	PMOR0120H
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION

☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action

☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.

Owner	Percent Interest
Jenam Tech, LLC	100%

The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)

9923995

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:

- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

☐ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.

- ☒ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.

THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

- ☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application
- Registration Number 44580
- ☐ A sole inventor
- ☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application
- ☐ A joint inventor; all of whom are signing this request

Signature	/Patrick Caldwell/
Name	Patrick Caldwell

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

EXHIBIT 28



UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/368,811	03/28/2019	Robert Paul Morris	PMOR0120I	8099
92045	7590	08/12/2019	EXAMINER	
The Caldwell Firm, LLC			MEKY, MOUSTAFA M	
PO Box 59655				
Dept. SVIPGP				
Dallas, TX 75229				
			ART UNIT	PAPER NUMBER
			2457	
			NOTIFICATION DATE	DELIVERY MODE
			08/12/2019	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eofficeaction@apcoll.com
pcaldwell@thecaldwellfirm.com

Office Action Summary**Application No.**

16/368,811

Applicant(s)

Morris, Robert Paul

Examiner

Moustafa M Meky

Art Unit

2457

AIA (FITF) Status

No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on 03/28/2019.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on ____.

2a) ☐ This action is **FINAL**.

2b) ☒ This action is non-final.

3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.

4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5) ☒ Claim(s) 1-20 is/are pending in the application.

5a) Of the above claim(s) ____ is/are withdrawn from consideration.

6) ☐ Claim(s) ____ is/are allowed.

7) ☒ Claim(s) 1-20 is/are rejected.

8) ☐ Claim(s) ____ is/are objected to.

9) ☐ Claim(s) ____ are subject to restriction and/or election requirement

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10) ☐ The specification is objected to by the Examiner.

11) ☒ The drawing(s) filed on 03/28/2019 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a) ☐ All b) ☐ Some** c) ☐ None of the:

1. ☐ Certified copies of the priority documents have been received.

2. ☐ Certified copies of the priority documents have been received in Application No. ____.

3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

3) ☐ Interview Summary (PTO-413)

Paper No(s)/Mail Date ____.

2) ☒ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)

4) ☐ Other: ____.

Paper No(s)/Mail Date 03/28/2019.

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

1. Claims 1-20 are presenting for examination.

Obvious Double Patent Rejection

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on nonstatutory double patenting provided the reference application or patent either is shown to be commonly owned with the examined application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement. See MPEP § 717.02 for applications subject to examination under the first inventor to file provisions of the AIA as explained in MPEP § 2159. See MPEP §§ 706.02(l)(1) - 706.02(l)(3) for applications not subject to examination under the

Application/Control Number: 16/368,811
Art Unit: 2457

Page 3

first inventor to file provisions of the AIA. A terminal disclaimer must be signed in compliance with 37 CFR 1.321(b).

The USPTO Internet website contains terminal disclaimer forms which may be used. Please visit www.uspto.gov/patent/patents-forms. The filing date of the application in which the form is filed determines what form (e.g., PTO/SB/25, PTO/SB/26, PTO/AIA/25, or PTO/AIA/26) should be used. A web-based eTerminal Disclaimer may be filled out completely online using web-screens. An eTerminal Disclaimer that meets all requirements is auto-processed and approved immediately upon submission. For more information about eTerminal Disclaimers, refer to www.uspto.gov/patents/process/file/efs/guidance/eTD-info-I.jsp.

3. Claims 1-2 are rejected on the ground of nonstatutory double patenting as being unpatentable over claims 1-2 of U.S. Patent No.8,219,606. Although the claims at issue are not identical, they are not patentably distinct from each other because claims 1-2 of the patent 606 teach substantially teach claims 1-2 of the pending application.

4. Claims 1-20 are rejected on the ground of nonstatutory double patenting as being unpatentable over claims 1-14 & 25-30 of U.S. Patent No.9,923,995. Although the claims at issue are not identical, they are not patentably distinct from each other because the patent 995 teach substantially teach the claims of the pending application.

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Moustafa M Meky whose telephone number is (571)272-4005. The examiner can normally be reached on Monday-Friday 9AM-5PM.

Application/Control Number: 16/368,811
Art Unit: 2457

Page 4

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on 571-272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/MOUSTAFA M MEKY/
Primary Examiner, Art Unit 2457
08/07/2019

Doc Code: DIST.E.FILE**Document Description: Electronic Terminal Disclaimer - Filed**U.S. Patent and Trademark Office
Department of Commerce

Electronic Petition Request	TERMINAL DISCLAIMER TO OBIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT
Application Number	16368811
Filing Date	28-Mar-2019
First Named Inventor	Robert Morris
Attorney Docket Number	PMOR0120I
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION

☒ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action

☒ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.

Owner	Percent Interest
Jenam Tech, LLC of Longview, TX	100%

The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)

9923995

8219606

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:

- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

- ☐ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.
- ☒ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.

THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

- ☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application
- Registration Number 44580
- ☐ A sole inventor
- ☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application
- ☐ A joint inventor; all of whom are signing this request

Signature	/Patrick Caldwell/
Name	Patrick Caldwell

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

EXHIBIT 29



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/079,397	10/23/2020	Robert Paul Morris	PMOR0120K	9756
92045	7590	12/08/2020	EXAMINER	
The Caldwell Firm, LLC			ALI, SYED	
PO Box 59655				
Dept. SVIPGP				
Dallas, TX 75229				
			ART UNIT	PAPER NUMBER
			2468	
			NOTIFICATION DATE	DELIVERY MODE
			12/08/2020	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eofficeaction@appcoll.com
pcaldwell@thecaldwellfirm.com

Office Action Summary**Application No.**

17/079,397

Applicant(s)

Morris, Robert Paul

Examiner

SYED ALI

Art Unit

2468

AIA (FITF) Status

No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on 10/23/2020.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on ____.

2a) ☐ This action is **FINAL**.

2b) ☒ This action is non-final.

3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.

4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5) ☒ Claim(s) 1-20 is/are pending in the application.

5a) Of the above claim(s) ____ is/are withdrawn from consideration.

6) ☐ Claim(s) ____ is/are allowed.

7) ☒ Claim(s) 1-20 is/are rejected.

8) ☐ Claim(s) ____ is/are objected to.

9) ☐ Claim(s) ____ are subject to restriction and/or election requirement

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10) ☐ The specification is objected to by the Examiner.

11) ☒ The drawing(s) filed on 10/23/2020 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a) ☐ All b) ☐ Some** c) ☐ None of the:

1. ☐ Certified copies of the priority documents have been received.

2. ☐ Certified copies of the priority documents have been received in Application No. ____.

3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

3) ☐ Interview Summary (PTO-413)

Paper No(s)/Mail Date ____.

2) ☐ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b)

4) ☐ Other: ____.

Paper No(s)/Mail Date ____.

DETAILED ACTION

Notice of Pre-AIA or AIA Status

The present application is being examined under the pre-AIA first to invent provisions.

1. This action is in response to the application filed on October 23, 2020.
2. Claims 1- 20 are under examination.

Double Patenting

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on nonstatutory double patenting provided the reference application or patent either is shown to be commonly owned with the examined application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement. See MPEP § 717.02 for applications subject to examination under the first inventor to file provisions of the AIA as explained in MPEP § 2159. See MPEP §§ 706.02(l)(1) - 706.02(l)(3) for applications not subject to examination under the first inventor to file provisions of the AIA. A terminal disclaimer must be signed in compliance with 37 CFR 1.321(b).

The USPTO Internet website contains terminal disclaimer forms which may be used. Please visit www.uspto.gov/patent/patents-forms. The filing date of the application in which the form is filed determines what form (e.g., PTO/SB/25, PTO/SB/26, PTO/AIA/25, or PTO/AIA/26) should be used. A web-based eTerminal Disclaimer may be filled out completely online using web-screens. An eTerminal Disclaimer that meets all requirements is auto-processed and approved immediately upon submission. For more information about eTerminal Disclaimers, refer to www.uspto.gov/patents/process/file/efs/guidance/eTD-info-I.jsp.

Claim 1 is rejected on the ground of nonstatutory double patenting as being unpatentable over claims 1-20 of U.S. Co-pending No. 16/914267. Although the claims at issue are not identical, they are not patentably distinct from each other because both the instant claims and the conflicting claims are directed to weighting matrices in a wireless communication device applications.

Conflicting Claim 1: *An apparatus comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for:*

receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established;

detecting an idle time period parameter field in the TCP-variant packet;
identifying metadata in the idle time period parameter field for an idle time period that is detectable by the first node and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active; and

modifying, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection.

Instant Claim 1: 1. *An apparatus comprising: a non-transitory memory storing instructions; and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the instructions for:*

receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established;

detecting an idle time period parameter field in the TCP-variant packet;
identifying metadata in the idle time period parameter field for an idle time period that is detectable by the first node and, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active; and

modifying, by the second node and based on the metadata, a timeout attribute associated with the TCP-variant connection.

The instant claim merely broadens the scope of the conflicting claims while adding a control unit. It is well settled that broadening the scope of claims would have been obvious to one of ordinary skill in the art in view of the narrower issued claims. In re Van Ornum, 686 F.2d 937, 214 USPQ 761 (CCPA 1982) and In re Goodman, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993).

Claim Rejections - 35 USC § 103

In the event the determination of the status of the application as subject to AIA 35 U.S.C. 102 and 103 (or as subject to pre-AIA 35 U.S.C. 102 and 103) is incorrect, any correction of the statutory basis for the rejection will not be considered a new ground of rejection if the prior art relied upon, and the rationale supporting the rejection, would be the same under either status.

This application currently names joint inventors. In considering patentability of the claims the examiner presumes that the subject matter of the various claims was commonly owned as of the effective filing date of the claimed invention(s) absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and effective filing dates of each claim that was not commonly owned as of the effective filing date of the later invention in order for the examiner to consider the applicability of 35 U.S.C. 102(b)(2)(C) for any potential 35 U.S.C. 102(a)(2) prior art against the later invention.

The following is a quotation of 35 U.S.C. 103 which forms the basis for all obviousness rejections set forth in this Office action:

Application/Control Number: 17/079,397

Page 6

Art Unit: 2468

A patent for a claimed invention may not be obtained, notwithstanding that the claimed invention is not identically disclosed as set forth in section 102, if the differences between the claimed invention and the prior art are such that the claimed invention as a whole would have been obvious before the effective filing date of the claimed invention to a person having ordinary skill in the art to which the claimed invention pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103 are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 1-20 is/are rejected under 35 U.S.C. 103 as being unpatentable over Sillasto et al. (USP 2005/0063304), in view of Appanna et al. (USP:2006/0062142).

As per Claim 1 Sillasto teaches an apparatus comprising:
a non-transitory memory storing instructions (**Paragraph 0093, 0113 the first packet(s) arrive at the RLC/PDCP buffer**); and
one or more processors in communication with the non-transitory memory,
wherein the one or more processors execute the instructions for (**Paragraph 0113 the UTRAN the buffering occurs in the RNC, and in the IP-BTS of the IP-RAN**):
receiving, by a second node from a first node, a transmission control protocol (TCP)-variant packet in advance of a TCP-variant connection being established

(Paragraph 0040, 0052 This message has a FIN flag, and it is one of the ending messages of a TCP connection. Each side of the TCP connection ends one direction of the TCP connection);

detecting an idle time period parameter field in the TCP-variant packet

(Paragraph 0040, 0051-0055 Inactivity timer is set on. There may be some delay between the actual detection of the emptiness of the buffer and the indication);

identifying metadata in the idle time period parameter field for an idle time period that is detectable by the first node and ***(Paragraph 0118 lengths of the inactivity and the activity periods (in time))***, during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active ***(Paragraph 0040, 0052, 0066, 0067 The inactivity timer is not set/reset when this small packet is sent.);*** and

modifying, by the second node and ***(Paragraph 0076, 0084 User data arrives after the FIN flag detection. The inactivity timer value may/shall be modified. This indicates that even if one TCP connection has terminated, there is/are one/several TCP connection(s) still on.).***

However, Sillasto does not explicitly disclose based on the metadata, a timeout attribute associated with the TCP-variant connection

Appanna disclose based on the metadata, a timeout attribute associated with the TCP-variant connection ***(Paragraph 0042The TCP metadata initially comprises information about TCP packet sizes, and after the TCP packet is acknowledged, information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep***

alive the TCP session as determined by, for example, a retransmission time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment, the standby transport module 230b generates dummy TCP packets based on the stateful metadata.)

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (see Appanna Paragraph 0042).

As per Claim 2 Sillasto – Appanna teaches the apparatus of claim 1 However, Sillasto does not explicitly disclose wherein the apparatus is configured such that the timeout attribute is an attribute of a keep-alive.

Appanna disclose wherein the apparatus is configured such that the timeout attribute is an attribute of a keep-alive (***Paragraph 0042The TCP metadata initially comprises information about TCP packet sizes, and after the TCP packet is acknowledged, information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by, for example, a retransmission***

time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment, the standby transport module 230b generates dummy TCP packets based on the stateful metadata.)

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (see *Appanna Paragraph 0042*).

As per Claim 3 Sillasto – Appanna teaches the apparatus of claim 1 wherein at least one of:

the second node includes a server, the server being configured to: in response to the receiving, send, by the second node to the first node, another TCP-variant packet in advance of the TCP-variant connection being established, the another TCP-variant packet including other metadata for the idle time period; or

the second node includes a client, the client being configured such that the receiving is performed subsequent to sending, by the second node to the first node, another TCP-variant packet in advance of the TCP-variant connection being established including other metadata for the idle time period (***Paragraph 0109 FIG. 12 comprises***

two radio access networks: the UTRAN and the IP-RAN. The IP-RAN (Internet Protocol Radio Access Network) is an RAN architecture that is fully optimised to carry IP traffic and is based on IP transport technology).

As per Claim 4 Sillasto – Appanna teaches the apparatus of claim 3 wherein, regardless as to whether the apparatus is the server or the client, the metadata is the same as the other metadata ***(Paragraph 0030 A TCP connection establishment is done on common transport channels (three way handshake with headers only i.e. very small packets). A dedicated transport channel (DCH) is allocated when actual data transmission starts.).***

However, Sillasto does not explicitly disclose based on the metadata, a timeout attribute associated with the TCP-variant connection

Appanna disclose based on the metadata, a timeout attribute associated with the TCP-variant connection ***(Paragraph 0042The TCP metadata initially comprises information about TCP packet sizes, and after the TCP packet is acknowledged, information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by, for example, a retransmission time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment, the standby transport module 230b generates dummy TCP packets based on the stateful metadata.)***

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (*see Appanna Paragraph 0042*).

As per Claim 5 Sillasto – Appanna teaches the apparatus of claim 3 wherein, regardless as to whether the apparatus is the server or the client, the metadata is different from the other metadata (***Paragraph 0035 a small packet arrival at a buffer is that a new session is initiated. Therefore, the inactivity timer value is set to the initial value***).

However, Sillasto does not explicitly disclose based on the metadata, a timeout attribute associated with the TCP-variant connection

Appanna disclose based on the metadata, a timeout attribute associated with the TCP-variant connection (***Paragraph 0042The TCP metadata initially comprises information about TCP packet sizes, and after the TCP packet is acknowledged, information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by, for example, a retransmission time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment,***

the standby transport module 230b generates dummy TCP packets based on the stateful metadata.)

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (see *Appanna Paragraph 0042*).

As per Claim 6 Sillasto – Appanna teaches the apparatus of claim 1 wherein the apparatus is configured such that the timeout attribute is specified in a number of seconds or minutes (***Paragraph 0113 This gives more information for the timer value decision. If for example the buffer has been loaded for some time, for example last five seconds there has been more than five packets all the time in buffer***).

However, Sillasto does not explicitly disclose based on the metadata, a timeout attribute associated with the TCP-variant connection

Appanna disclose based on the metadata, a timeout attribute associated with the TCP-variant connection (***Paragraph 0042The TCP metadata initially comprises information about TCP packet sizes, and after the TCP packet is acknowledged, information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep***

alive the TCP session as determined by, for example, a retransmission time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment, the standby transport module 230b generates dummy TCP packets based on the stateful metadata.)

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (see Appanna Paragraph 0042).

As per Claim 7 Sillasto – Appanna teaches the apparatus of claim 1 However, Sillasto does not explicitly disclose wherein the apparatus is configured such that the timeout attribute is used to keep the TCP-variant connection open when inactive, and to prevent one or more other nodes from closing the TCP-variant connection when inactive.

Appanna disclose wherein the apparatus is configured such that the timeout attribute is used to keep the TCP-variant connection open when inactive, and to prevent one or more other nodes from closing the TCP-variant connection when inactive.
(Paragraph 0042The TCP metadata initially comprises information about TCP

packet sizes, and after the TCP packet is acknowledged, information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by, for example, a retransmission time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment, the standby transport module 230b generates dummy TCP packets based on the stateful metadata.)

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (see *Appanna Paragraph 0042*).

As per Claim 8 Sillasto – Appanna teaches the apparatus of claim 1 wherein the apparatus is configured such that the metadata is used as input of an algorithm for determining a duration of time specified by the timeout attribute (***Paragraph 0007 An inactivity timer is a timer which sets the maximum duration of a DCH (Dedicated CHannel) allocation after data transfer has ceased.***).

As per Claim 9 Sillasto – Appanna teaches the apparatus of claim 8 wherein the apparatus is configured such that the algorithm is determined based on at least one particular attribute ***(Paragraph 0030, 0035 the inactivity timer has higher value (10) since interruptions during the transmission occur because of the TCP slow start algorithm.).***

As per Claim 10 Sillasto – Appanna teaches the apparatus of claim 1 wherein the apparatus is configured such that the modification of the timeout attribute results from a negotiation between the first node and the second node via a negotiation protocol of a TCP-variant protocol ***(Paragraph 0013 The present UTRAN and IP-RAN comprise release (inactivity) timers for the NRT bearers).***

As per Claim 11 Sillasto – Appanna teaches the apparatus of claim 1 wherein the one or more processors execute the instructions for:

and in response to detecting the idle time period, deactivating the TCP-variant connection by releasing a resource allocated for the TCP-variant connection by one of the first and second nodes without signaling another one of the first and second nodes that is related to the detection of the idle time period ***(Paragraph 0007, 0030 An inactivity timer is a timer which sets the maximum duration of a DCH (Dedicated CHannel) allocation after data transfer has ceased).***

However, Sillasto does not explicitly disclose detecting the idle time period based on the timeout attribute;

Appanna disclose detecting the idle time period based on the timeout attribute

(Paragraph 0042The TCP metadata initially comprises information about TCP packet sizes, and after the TCP packet is acknowledged, information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by, for example, a retransmission time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment, the standby transport module 230b generates dummy TCP packets based on the stateful metadata.)

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to the make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (see Appanna Paragraph 0042).

As per Claim 12 Sillasto – Appanna teaches the apparatus of claim 1 wherein the apparatus is configured such that at least one of the detecting or the identifying is performed at a TCP-variant layer other than a

TCP layer, where the TCP-variant layer is above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer (**Paragraph 0030, 0071 FIG. 1 illustrates an example of an HTTP/TCP session (HTTP, Hyper Text Transport Protocol). A TCP connection establishment is done on common transport channels (three way handshake with headers only i.e. very small packets).**

As per Claim 13 Sillasto – Appanna teaches the apparatus of claim 1 wherein the one or more processors execute a network application that is configured to perform a 3-way TCP handshake for establishing a TCP connection that is different than the TCP-variant connection, and wherein the network application is configured to establish the TCP-variant connection instead of the TCP connection in order to permit negotiation, between the first node and the second node, of the timeout attribute, where the timeout attribute is not available when establishing the TCP connection, but is available when establishing the TCP-variant connection so that the TCP-variant connection is capable of being at least partially closed when inactive based on (**Paragraph 0030, 0035 A TCP connection establishment is done on common transport channels (three way handshake with headers only i.e. very small packets). A TCP connection establishment is done on common transport channels (three way handshake with headers only i.e. very small packets).**

However, Sillasto does not explicitly disclose based on the timeout attribute;

Appanna disclose based on the timeout attribute (**Paragraph 0042The TCP metadata initially comprises information about TCP packet sizes, and after the**

TCP packet is acknowledged, information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by, for example, a retransmission time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment, the standby transport module 230b generates dummy TCP packets based on the stateful metadata.)

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (see Appanna Paragraph 0042).

As per Claim 14 Sillasto – Appanna teaches the apparatus of claim 1 wherein the apparatus is configured such that the TCP-variant packet and the metadata included therewith are received by the second node from the first node, without any prior signaling from the second node to the first node (***Paragraph 0109 FIG. 12 comprises two radio access networks: the UTRAN and the IP-RAN***).

As per Claim 15 Sillasto – Appanna teaches an apparatus comprising:

a non-transitory memory storing a network application (***Paragraph 0113 the UTRAN the buffering occurs in the RNC, and in the IP-BTS of the IP-RAN***); and one or more processors in communication with the non-transitory memory, wherein the one or more processors execute the network application such that the network application is configured to operate in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer (***Paragraph 0030, 0071 FIG. 1 illustrates an example of an HTTP/TCP session (HTTP, Hyper Text Transport Protocol). A TCP connection establishment is done on common transport channels (three way handshake with headers only i.e. very small packets)),*** the apparatus, when operating in accordance with the non-TCP protocol, configured to (***Paragraph 0030, 0109, 0113 This is clearly different from just using IP as a transport solution with the existing network architectures like the GSM (Global System for Mobile Communications) and The radio access networks are connected to the core network CN.):***

receive, by a second node from a first node, a non-TCP packet during a setup of a non-TCP connection (***Paragraph 0007, 0052 This message has a FIN flag, and it is one of the ending messages of a TCP connection. The Forward access channel (FACH) is a downlink transport channel that is used to carry control information from the base station to the mobile station.***);

identify metadata, that specifies a number of seconds or minutes, in an idle time period parameter field in the non-TCP packet, for an idle time period that is detectable by the first node, where, as a result of a detection of the idle time period, the non-TCP

connection is subject to deactivation (***Paragraph 0007,0118 lengths of the inactivity and the activity periods (in time) The Forward access channel (FACH) is a downlink transport channel that is used to carry control information from the base station to the mobile station.***); and

determine, based on the metadata, a timeout attribute associated with the non-TCP connection;

wherein the apparatus, when operating in accordance with the TCP protocol, is configured to perform a three-way TCP handshake for establishing a TCP connection that is different than the non-TCP connection (***Paragraph 0030, 0035, 0071 A TCP connection establishment is done on common transport channels (three way handshake with headers only i.e. very small packets).***).

However, Sillasto does not explicitly disclose based on the metadata, a timeout attribute associated with the TCP-variant connection

Appanna disclose based on the metadata, a timeout attribute associated with the TCP-variant connection (***Paragraph 0042The TCP metadata initially comprises information about TCP packet sizes, and after the TCP packet is acknowledged, information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by, for example, a retransmission time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment, the standby transport module 230b generates dummy TCP packets based on the stateful metadata.***)

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (*see Appanna Paragraph 0042*).

As per Claim 16 Sillasto – Appanna teaches the apparatus of claim 15 wherein the apparatus is configured such that:

the non-TCP packet and the idle time period parameter field included therewith are operable for being received by the second node from the first node, without any prior signaling from the second node to the first node **(Paragraph 0040, 0052 the UL messages are monitored, and the SYN flag in the TCP header is detected, this triggers the clearance of the inactivity timer. Also the DL inactivity timer can be cleared when a SYN flag is detected in UL direction, and vice versa.**

The inactivity timer value may be changed for a new TCP session);

the determination of the timeout attribute results from a negotiation between the first node and the second node; and

the metadata includes a first value and the timeout attribute is determined to include a second value that is different than the first value of the metadata.

However, Sillasto does not explicitly disclose based on the metadata, a timeout attribute associated with the TCP-variant connection

Appanna disclose based on the metadata, a timeout attribute associated with the TCP-variant connection ***(Paragraph 0042The TCP metadata initially comprises information about TCP packet sizes, and after the TCP packet is acknowledged, information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by, for example, a retransmission time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment, the standby transport module 230b generates dummy TCP packets based on the stateful metadata.)***

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to the make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (see Appanna Paragraph 0042).

As per Claim 17 Sillasto – Appanna teaches the apparatus of claim 15 However, Sillasto does not explicitly disclose wherein the apparatus is configured for: detecting

the idle time period based on the timeout attribute; and deactivating the non-TCP connection by communicating a particular packet between the first node and the second node, in response to detecting the idle time period

Appanna disclose wherein the apparatus is configured for: detecting the idle time period based on the timeout attribute; and deactivating the non-TCP connection by communicating a particular packet between the first node and the second node, in response to detecting the idle time period **(Paragraph 0042The TCP metadata initially comprises information about TCP packet sizes, and after the TCP packet is acknowledged, information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by, for example, a retransmission time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment, the standby transport module 230b generates dummy TCP packets based on the stateful metadata.)**

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to the make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (see Appanna Paragraph 0042).

As per Claim 18 Sillasto teaches an apparatus comprising:

a non-transitory memory storing a network application; and one or more processors in communication with the non-transitory memory (**Paragraph 0113 the UTRAN the buffering occurs in the RNC, and in the IP-BTS of the IP-RAN**), wherein the one or more processors execute the network application such that the network application is configured to operate in accordance with a non-transmission control protocol (TCP) protocol that operates above an Internet Protocol (IP) layer and below a hypertext transfer protocol (HTTP) application layer, the apparatus, when operating in accordance with the non-TCP protocol, configured to (**Paragraph 0030, 0071 FIG. 1 illustrates an example of an HTTP/TCP session (HTTP, Hyper Text Transport Protocol). A TCP connection establishment is done on common transport channels (three way handshake with headers only i.e. very small packets)),** the apparatus, when operating in accordance with the non-TCP protocol, configured to (**Paragraph 0030, 0109, 0113 This is clearly different from just using IP as a transport solution with the existing network architectures like the GSM (Global System for Mobile Communications) and The radio access networks are connected to the core network CN.):**

receive idle information for use in detecting an idle time period that results in a non-TCP connection being subject to deactivation (**Paragraph 0007, 0052 This message has a FIN flag, and it is one of the ending messages of a TCP connection. The Forward access channel (FACH) is a downlink transport channel**

that is used to carry control information from the base station to the mobile station.);

generate, based on the idle information, a non-TCP packet including an idle time period parameter field identifying metadata that is specified in a number of seconds or minutes; ***(Paragraph 0007, 0118 lengths of the inactivity and the activity periods (in time) The Forward access channel (FACH) is a downlink transport channel that is used to carry control information from the base station to the mobile station.);*** and

send, from a first node to a second node and for establishing the non-TCP connection, the non-TCP packet to provide the metadata to the second node, for use by the second node in determining a timeout attribute associated with the non-TCP connection ***(Paragraph 0071, 0109 FIG. 12 comprises two radio access networks: the UTRAN and the IP-RAN);***

wherein the apparatus, when operating in accordance with the TCP protocol, is configured to perform a three-way TCP handshake for establishing a TCP connection that is separate from the non-TCP connection ***(Paragraph 0030, 0035, 0071 A TCP connection establishment is done on common transport channels (three way handshake with headers only i.e. very small packets).).***

However, Sillasto does not explicitly disclose based on the metadata, a timeout attribute associated with the TCP-variant connection

Appanna disclose based on the metadata, a timeout attribute associated with the TCP-variant connection ***(Paragraph 0042 The TCP metadata initially comprises information about TCP packet sizes, and after the TCP packet is acknowledged,***

information about the ACK. When a stateful switchover occurs, it is the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by, for example, a retransmission time out or a fast retransmission algorithm. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. In another embodiment, the standby transport module 230b generates dummy TCP packets based on the stateful metadata.)

It would have been obvious to one of ordinary skill in the art before the effective filing date of the claimed invention to modify Sillasto with the teachings of Appanna in order to make the system more efficient. Because the modification would enable the TCP packets that have yet to be acknowledged that may need to be resent to keep alive the TCP session as determined by a retransmission time out. Accordingly, stateful metadata comprises that portion of metadata needed to maintain the TCP session. (see Appanna Paragraph 0042).

As per Claim 19 Sillasto – Appanna teaches the apparatus of claim 18 wherein the apparatus is configured such that:

the determination of the timeout attribute results from a negotiation between the first node and the second node (***Paragraph 0030, 0035 A TCP connection establishment is done on common transport channels (three way handshake with headers only i.e. very small packets). A TCP connection establishment is done on***

common transport channels (three way handshake with headers only i.e. very small packets));

during the idle time period, no non-TCP packet including data is communicated in the non-TCP connection ***(Paragraph 0007, 0052 This message has a FIN flag, and it is one of the ending messages of a TCP connection. The Forward access channel (FACH) is a downlink transport channel that is used to carry control information from the base station to the mobile station.);***

wherein the apparatus is further configured for:

detecting the idle time period based on the idle information; and in response to detecting the idle time period, deactivating the non-TCP connection by at least partially closing the TCP-variant connection by one of the first and second nodes without communication between the second node and the first node that is related to the detection of the idle time period ***(Paragraph 0007, 0030 An inactivity timer is a timer which sets the maximum duration of a DCH (Dedicated CHannel) allocation after data transfer has ceased).***

As per Claim 20 Sillasto – Appanna teaches the apparatus of claim 19 wherein at least one of: the apparatus is at least one component of the first node;

the communication includes only receiving;

the communication includes only sending;

the communication includes receiving and sending;

the non-TCP protocol includes a variant to the TCP;

the non-TCP packet is sent directly from the first node to the second node;

the non-TCP packet is sent from the first node to the second node via at least one other node;

the idle time period parameter field is part of a data portion in the non-TCP packet;

the idle time period parameter field is part of a header of the non-TCP packet;

the idle information is received based on a previous header;

the timeout attribute is capable being the same as the metadata;

the timeout attribute is capable being different from the metadata;

the idle time period is specified in a number of seconds;

the idle time period is specified in a number of minutes;

the timeout attribute specifies a duration;

the non-TCP packet is informational;

the TCP protocol operates directly above the IP layer;

the TCP protocol operates directly below the HTTP application layer;

the non-TCP protocol operates directly above the IP layer;

the non-TCP protocol operates directly below the HTTP application layer;

the determination of the timeout attribute includes at least one of modifying, creating, or deleting the timeout attribute;

the network application provides the idle information to a ITP policy component via a settings service component interoperating with a sockets component;

during the idle time period, a non-TCP packet including data is received by the first node where the non-TCP packet including data is sent by another node; or

during the idle time period, a non-TCP packet including data is received by the first node where the non-TCP packet including data is sent by the second node in the TCP connection (***Paragraph 0071, 0109 FIG. 12 comprises two radio access networks: the UTRAN and the IP-RAN***).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to SYED ALI whose telephone number is (571)270-3681. The examiner can normally be reached on M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Asad Nawaz can be reached on (571)272-3988. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/SYED ALI/
Examiner, Art Unit 2468

Doc Code: DIST.E.FILE Document Description: Electronic Terminal Disclaimer - Filed		PTO/SB/25 PTO/SB/26 U.S. Patent and Trademark Office Department of Commerce
Electronic Petition Request	TERMINAL DISCLAIMER TO OBVIATE A PROVISIONAL DOUBLE PATENTING REJECTION OVER A PENDING "REFERENCE" APPLICATION AND TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT	
Application Number	17079397	
Filing Date	23-Oct-2020	
First Named Inventor	Robert Morris	
Attorney Docket Number	PMOR0120K	
Title of Invention	METHODS, SYSTEMS, AND COMPUTER PROGRAM PRODUCTS FOR SHARING INFORMATION FOR DETECTING AN IDLE TCP CONNECTION	
<input checked="" type="checkbox"/> Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action <input checked="" type="checkbox"/> This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.		
Owner	Percent Interest	
Jenam Tech, LLC	100 %	
The owner(s) of percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of any patent granted on pending reference Application Number(s)		
16914267 filed on 06/26/2020 13477402 filed on 05/22/2012 as the term of any patent granted on said reference application may be shortened by any terminal disclaimer filed prior to the grant of any patent on the pending reference application. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and any patent granted on the reference application are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.		
In making the above disclaimer, the owner does not disclaim the terminal part of any patent granted on the instant application that would extend to the expiration date of the full statutory term of any patent granted on said reference application, "as the term of any patent granted on said reference application may be shortened by any terminal disclaimer filed prior to the grant of any patent on the pending reference application," in the event that any such patent granted on the pending reference application: expires for failure to pay a maintenance fee, is held unenforceable, is found invalid by a court of competent jurisdiction, is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321, has all claims canceled by a reexamination certificate, is reissued, or is in any manner terminated prior to the expiration of its full statutory term as shortened by any terminal disclaimer filed prior to its grant.		

The owner(s) with percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)

9923995

10075564

10375215

10742774

8219606

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:

- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

☒ Terminal disclaimer fee under 37 CFR 1.20(d) is included with Electronic Terminal Disclaimer request.

☐ I certify, in accordance with 37 CFR 1.4(d)(4), that the terminal disclaimer fee under 37 CFR 1.20(d) required for this terminal disclaimer has already been paid in the above-identified application.

Applicants claims the following fee status:

☐ Small Entity

☐ Micro Entity

☒ Regular Undiscounted

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

THIS PORTION MUST BE COMPLETED BY THE SIGNATORY OR SIGNATORIES

I certify, in accordance with 37 CFR 1.4(d)(4) that I am:

- ☒ An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application
- Registration Number 44580
- ☐ A sole inventor
- ☐ A joint inventor; I certify that I am authorized to sign this submission on behalf of all of the inventors as evidenced by the power of attorney in the application
- ☐ A joint inventor; all of whom are signing this request

Signature	/Patrick Caldwell/
Name	Patrick Caldwell

*Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner).
Form PTO/SB/96 may be used for making this certification. See MPEP § 324.

EXHIBIT 30

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC, LG ELECTRONICS, INC.,
LG ELECTRONICS U.S.A., INC.,

Petitioners

v.

JENAM TECH, LLC,

Patent Owner

Case IPR2020-00845
U.S. Patent No. 9,923,995

PATENT OWNER'S REQUEST FOR REHEARING

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
Submitted Electronically via PTAB E2E

TABLE OF CONTENTS

I. INTRODUCTION1

II. THE BOARD MISAPPREHENDED PATENT OWNER’S CLAIM
CONSTRUCTION2

 A. The Board Adopted, But Misapprehended and Misapplied, Patent
 Owner’s Claim Construction2

 B. The Board’s Misapprehension of Patent Owner’s Claim Construction
 Tainted its Invalidity Analysis.....3

 1. The Board Overlooked the “With or Without a Condition”
 Requirement of Patent Owner’s Construction.....3

 2. The Board Erroneously Imported a “Sufficiency” Requirement into
 the Construction of the ITP Limitation.....8

III. CONCLUSION.....11

TABLE OF AUTHORITIES

Cases	Page(s)
<i>Amazon.com, Inc. v. Barnesandnoble.com, Inc.</i> , 239 F.3d 1343 (Fed. Cir. 2001)	10

I. INTRODUCTION

Pursuant to 37 C.F.R. § 42.71, Patent Owner Jenam Tech, LLC respectfully requests rehearing of the Board’s Final Written Decision (“FWD”) dated August 27, 2021.

Here, the Board misapprehended Patent Owner’s proposed claim construction of the idle time period (“ITP”) limitation in the challenged claims; specifically, the sub-limitation “to keep the TCP-variant connection active.” In the FWD, the Board did not dispute Patent Owner’s proposal that the sub-limitation be construed to mean “to keep the TCP-variant connection active with or without a condition and via any mechanism,” and purported to adopt it. (FWD at 8-9.) However, the Board subsequently overlooked the “with or without a condition” requirement of the construction. In other words, as previously noted,¹ it misapprehended that a packet which is communicated to attempt to keep the connection active, whether *directly* (*i.e.*, without a condition) or *indirectly* (*i.e.*, with a condition), does not meet the “no packet . . .” limitation of the challenged claims.² The Board’s misapprehension of Patent Owner’s proposed construction, and its subsequent application of an improper construction were legal error.

¹ (PO’s Sur-Reply at 23.)

² Patent Owner’s case-in-chief relies on packets that always occur (e.g., *Eggert*’s retransmissions). To the extent Patent Owner previously referenced hypothetical scenarios that occur “sometimes,” in this or other parallel proceedings, Patent Owner rescinds all such arguments.

Patent Owner therefore requests rehearing and that the Board find that Petitioners have not met their burden with respect to any challenged claim. Patent Owner

II. THE BOARD MISAPPREHENDED PATENT OWNER’S CLAIM CONSTRUCTION

A. The Board Adopted, But Misapprehended and Misapplied, Patent Owner’s Claim Construction

In the FWD, the Board purported to adopt and apply Patent Owner’s construction regarding the phrase “to keep the TCP-variant connection active.” (FWD at 9 (“we apply Patent Owner’s construction of the phrase ‘to keep the TCP-variant connection active’ and need not resolve any disputes regarding the scope of this phrase.”).) Patent Owner’s construction, in pertinent part, is “to keep the TCP-variant connection active with or without a condition and via any mechanism.” (*Id.* at 8-9.) Stated differently, Patent Owner’s construction is that a packet which is communicated in an attempt to keep the connection active, whether directly (*i.e.*, without a condition) or indirectly (*i.e.*, with a condition), does not meet the “no packet . . .” limitation of the challenged claims. This is also true where the packet does not actually result in keeping the connection active, absent a condition (e.g., receipt of an elicited ACK via an intact physical connection). But the Board misapprehended these aspects of Patent Owner’s construction. (*Id.* at 9.) That error permeated the Board’s FWD and is sufficient to warrant rehearing.

B. The Board’s Misapprehension of Patent Owner’s Claim Construction Tainted its Invalidity Analysis

1. The Board Overlooked the “With or Without a Condition” Requirement of Patent Owner’s Construction

As Patent Owner explained, the phrase “with or without a condition” in its construction explicitly precludes packets that are communicated “to keep the TCP-variant connection active” even if a condition exists (*i.e.*, “with a condition” or, to put it differently, indirectly) before the result (keeping the connection active) may be achieved. (POR at 26 (citing Ex. 2020 at ¶61); PO Sur-Reply at 23.) In other words, Patent Owner’s construction clarifies that a packet is still “*to keep* the TCP-variant connection active” even if the connection is *not* ultimately kept alive due to an absence of a required “condition,” such as the receipt of an ACK via an intact physical connection.

However, in its invalidity analysis, the Board overlooked the above and concluded that retransmission packets in *Eggert* are not packets “to keep the TCP-variant connection active” as they involve a condition precedent before the connection can be kept active:

Eggert does not disclose, and Patent Owner does not argue, that retransmitting data resets the original timeout timer that started upon sending the data the first time. Thus, despite retransmitting the data, the connection will terminate on the expiration of the ATO timer that started when the first data were transmitted.

(FWD at 18.) However, this finding overlooks Patent Owner’s arguments under its construction that “retransmitting data resets the original timeout” with a condition,

where the “condition” is met, namely, with the receipt of an ACK that is elicited by such retransmitted packets. (PO Sur-Reply at 22-25.) That is, the retransmission packets are sent to keep the connection active, despite being subject to a condition or, in other words, despite being indirectly responsible for keeping the connection active. Moreover, if this “condition” (e.g., receipt of the responsive ACK) is *not* met, such as due to a disconnection, the claims are still not met by *Eggert* under Patent Owner’s construction—retransmission packets are still sent in an attempt to keep the connection active, albeit indirectly.

Instead of applying the foregoing, the Board misapprehended and misapplied Patent Owner’s construction by concluding that the ITP limitation is restricted to where “no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active” *without a condition*, (*i.e.*, directly) thereby improperly broadening the claims.

The Board confirmed this misinterpretation where the negative ITP limitation applies *only* to transmissions to keep the connection active *without* a condition, and applied it in its invalidity analysis. (FWD at 19 (“If the *purpose* of retransmitting the data were to keep the connection active, then the device would be programmed to reset the ATO timer upon retransmission to avoid termination of the connection.”).) There, the Board discounted packets sent to keep the connection active *with a condition, i.e., indirectly*.

Similarly, with this incorrect interpretation, the Board concluded that in order for the claims to be distinguished from *Eggert*, “the device would [need to] be programmed to reset the ATO timer upon retransmission,” regardless of the receipt of a responsive ACK. (FWD at 19.) In doing so, the Board limited the ITP limitation to packets sent to keep the connection active “without a condition.” However, the Board was incorrect in reasoning that the claim phrase at issue only applies to packets that, by themselves, *directly* avoid connection termination.

As summarized in Patent Owner’s Demonstrative Slides at 29, the ITP limitation cannot be construed to only encompass such a “without a condition” requirement as that would not only: (a) be inconsistent with how the Board previously applied such ITP limitation, and (b) improperly exclude disclosed embodiments, but it would also (c) violate the doctrine of claim differentiation. Specifically, claims 110, 111, and 114 of the related ’742 patent not only preclude importation of a “*without* a condition” or similar limitation, but also confirm that the overall ITP limitation is broad enough such that the *negative* “no packet” limitation properly excludes packets that keep the connection active “with a condition” (*i.e.*, indirectly):

110. The method of claim 104, wherein the idle time period is one: during which, no packet is communicated in the TCP-variant connection to keep the TCP-variant connection active *with a condition*.

111. The method of claim 104, wherein the idle time period is one: during which, no packet is communicated in the TCP-variant

connection to keep the TCP-variant connection active ***without a condition***.

114. The method of claim 104, wherein the idle time period is one: during which, no packet is communicated in the TCP-variant connection to ***unconditionally*** keep the TCP-variant connection active.

(See PO Sur-Reply at 13 (emphasis added).)

The Board’s reasoning overlooks these claim differentiation arguments previously advanced by Patent Owner.³ The table below summarizes Patent Owner’s proposed “with or without a condition” construction and demonstrates how it encompasses (1) situations where a packet is transmitted to keep the connection active “***without*** a condition” ***and*** (2) situations where a packet is transmitted to keep the connection active “***with*** a condition”.

Patent Owner’s “with or without a condition” construction that covers both “with a condition” AND “without a condition”	Separate components of Patent Owner’s “with <u>or</u> without a condition” construction	Packet communication covered by the separate components of Patent Owner’s “with <u>or</u> without a condition” construction
“idle time period and, during which, no packet is communicated in the	(1) “without a condition”	reset in direct response to a retransmission, to avoid termination of the

³ To the extent the Board read in a “sufficiency” requirement such that the intended result (*i.e.*, keeping the connection active) must actually be achieved, it also erred. Claims 112/113 of the related ’742 patent, recite that “the idle time period is one: during which, no packet is communicated in the TCP-variant connection that is ***sufficient to keep the TCP-variant connection active***” and that “the idle time period is one: during which, no packet is communicated in the TCP-variant connection to ***fulfill a purpose*** to keep the TCP-variant connection active.” (PO Sur-Reply at 23.) Thus, claim differentiation demonstrates that any such construction would be incorrect. (PO Sur-Reply at 13.)

TCP-variant connection to keep the TCP-variant connection active <i>with or without a condition</i>		connection (<i>i.e.</i> , no condition)
	(2) “with a condition”	reset <i>if</i> an ACK is received (<i>i.e.</i> , condition) in response to a retransmission, which is indirectly responsible for avoiding termination of the connection

As set forth above, the Board purportedly applied Patent Owner’s proposed “with or without a condition” construction but failed to recognize that retransmissions in *Eggert* serve to avoid termination of the connection “with a condition” [(2) above], namely when the elicited ACK is indeed received. The record is clear that ***but for*** “the ACK-eliciting retransmitted-packets,” no ACKs can be received in response to such retransmitted-packets, and the “connection will prematurely close without them.” (PO Sur-Reply at 23.)

As Patent Owner explained, the Board’s failure to apply the proposed “with or without a condition” construction not only “[c]ontradicts how the Board applied the same claims to different art in IPR No. 2020-00742,” (POR, 28-32), and “[e]xcludes multiple embodiments” (POR, 25-27), but also “[v]iolates claim differentiation” (PO Sur-Reply at 12-13; *see also* Patent Owner Demonstrative Slides at 29.) Conversely, under a proper application of Patent Owner’s proposed construction, the challenged claims are patentably distinct from *Eggert*’s

retransmission packets, which are communicated to “keep the TCP-variant connection active,” albeit “with . . . a condition” (e.g., receipt of an ACK that is elicited by the retransmission).

2. The Board Erroneously Imported a “Sufficiency” Requirement into the Construction of the ITP Limitation

Rather than properly apply the “with or without a condition” requirement of Patent Owner’s construction, the Board instead pivoted to relying on the “sufficiency” of the retransmitted packets in Eggert. (FWD at 19.) The Board’s approach was erroneous.

The Board’s error here is highlighted by the below annotations to its statements in the FWD referencing the “purpose” of Eggert’s retransmitted packets, which the Board equated with “sufficiency” rather than Patent Owner’s usage of “purposed.” That is, the Board misapprehended Patent Owner’s statements and instead equated the “purpose” with something that will actually be directly “sufficient” to achieve the intended result. (FWD at 19-20.) As seen below, the annotations illustrate the fact that the Board incorporated a “*without* a condition” limitation and overlooked Patent Owner’s actual proposal of “*with or without* a condition.”

If the purpose of retransmitting the data were to keep the connection active, then the device would be programmed to reset the ATO timer upon retransmission [*without a condition (such as receiving a responsive ACK)*] to avoid termination of the connection. The fact that this is not done supports a finding that the purpose of these packets is

not to keep the connection active [*without a condition (such as receiving a responsive ACK)*]

(FWD at 19.)

Thus, if the purpose of particular programming were to keep the connection active [*without a condition (such as receiving a responsive ACK)*], the device would be so programmed by resetting the ATO timer.

(FWD at 19.)

The possibility that a network condition such as a disconnection would prevent delivery of an ACK-eliciting packet is all the more reason that the ATO timer would be reset upon retransmission if the purpose of the retransmission packet indeed were to keep the connection active [*without a condition (such as receiving a responsive ACK)*].

(FWD at 20.)

In each of the above excerpts, the Board erroneously assumed that there is only one hypothetical way “to keep the TCP-variant connection active” via retransmissions, namely by resetting an ATO timeout in direct response to sending a retransmission, regardless of receipt of any responsive ACK. As shown in the annotations, the Board’s assumption improperly imports a “*without* a condition” limitation into the claims but overlooks the fact that under Patent Owner’s actual construction, it is required that no packets are communicated to keep the connection active even *with* a condition. (See POR at 22-25.) As Patent Owner previously explained, “Patent Owner purposefully did *not* claim *how* (e.g., with or without a condition) the packet is purposed “to keep the TCP-variant connection active.” (POR at 22.) Stated differently, a packet that is communicated to keep the

connection active, whether directly (*i.e.*, without a condition) or indirectly (*i.e.*, with a condition), does not meet the “no packet . . .” limitation of the challenged claims. As a result of its erroneous claim construction analysis, the Board failed to recognize that retransmissions in *Eggert*, which are admittedly intended to keep the connection active subject to a condition (e.g., receipt of a responsive ACK, that may or may not happen), are still “to keep the TCP-variant connection active.”

Moreover, the fact that the Board relied on the above hypotheticals regarding an alleged ATO timer⁴ in *Eggert* to conclude that retransmission of *data* was not sufficient for keeping the connection active demonstrates that the Board incorporated such a “sufficiency” requirement into the construction. That was incorrect. Indeed, the Board appears to have used the prior art itself as a guide to claim construction, which is legal error. *Amazon.com, Inc. v. Barnesandnoble.com, Inc.*, 239 F.3d 1343, 1351 (Fed. Cir. 2001) (“Only when a claim is properly understood can a determination be made whether . . . the prior art anticipates and/or renders obvious the claimed invention.”).

The Board purported to rely on a statement in the POR as support for its “sufficiency”-based analysis, (FWD at 9 (citing POR at 21)), but the Board again

⁴ Patent Owner does not concede nor admit that a separate “ATO timer” is disclosed in the cited references.

misapprehended Patent Owner’s positions. For instance, the Board overlooked other portions of the POR:

Patent Owner’s unconstrued claims are clearly silent as to the existence (or non-existence) of any condition(s) that must be satisfied before a particular packet ultimately results in keeping the connection active (e.g., by avoiding connection deactivation), and Patent Owner’s construction merely emphasizes and clarifies the same. (Ex. 2020 at 54-65.) To this end, as confirmed by Patent Owner’s construction, the unconstrued phrase “to keep the TCP-variant connection active” clearly describes a purpose of the “packet [that] is communicated” in the previous phrase, *and clearly does not require fulfillment of such purpose since such result may or may not necessarily come to fruition until one or more additional conditions are met.*

(POR at 25 (citing Ex. 2020 at 54-65).) Patent Owner previously explained that relying on the “purpose” of a transmitted packet would be futile, in the absence of additional conditions. Yet, because the Board construed this limitation to exclude any such conditions, the Board’s invalidity analysis based on the “sufficiency” of retransmission packets is mistaken and its ultimate conclusions are mistaken.

III. CONCLUSION

For these reasons, Patent Owner respectfully requests rehearing of the Board’s Final Written Decision dated August 27, 2021.

Dated: September 27, 2021

DEVLIN LAW FIRM LLC

/s/Timothy Devlin

Timothy Devlin (Reg. No. 41,706)

Derek Dahlgren (admitted *pro hac vice*)

1526 Gilpin Avenue

Wilmington, DE 19806

(302)-449-9010

tdevlin@devlinlawfirm.com

ddahlgren@devlinlawfirm.com

Attorneys for Patent Owner

CERTIFICATE OF SERVICE

I hereby certify that on September 27, 2021, I caused a true and correct copy of the foregoing materials to be served via the Patent Office electronic filing system, and electronic service via email to the following attorneys of record pursuant to Petitioners' consent.

<u>LEAD COUNSEL</u>	<u>BACK-UP COUNSEL</u>
Naveen Modi (Reg. 46,224) naveenmodi@paulhastings.com Paul Hastings LLP 2050 M. St. N.W. Washington, D.C. 20036 Telephone: (202) 551-1700 Additional email for service: PH-Google-Jenam- IPR@paulhastings.com	Joseph E. Palys (Reg. 46,508) josephpalys@paulhastings.com Quadeer A. Ahmed (Reg. 60,835) quadeerahmed@paulhastings.com Jason Heidemann (Reg. 77,880) jasonheidemann@paulhastings.com Paul Hastings LLP 2050 M. St. N.W. Washington, D.C. 20036 Telephone: (202) 551-1700

/s/ Timothy Devlin

Timothy Devlin